



UNICEUB - CENTRO UNIVERSITÁRIO DE BRASÍLIA

FAET – FACULDADE DE CIÊNCIAS EXATAS E TECNOLOGIA

CURSO DE ENGENHARIA DA COMPUTAÇÃO

RESTAURAÇÃO DE IMAGENS DIGITAIS UTILIZANDO FILTRAGEM INVERSA E FILTRAGEM WIENER

Aluna:

Fernanda Sakamoto Alves – RA: 9965846

Orientador:

Professor MC. Claudio Penedo de Albuquerque

Brasília, DF – fevereiro de 2005.



UNICEUB - CENTRO UNIVERSITÁRIO DE BRASÍLIA
FAET – FACULDADE DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE ENGENHARIA DA COMPUTAÇÃO

RESTAURAÇÃO DE IMAGENS DIGITAIS UTILIZANDO FILTRAGEM INVERSA E FILTRAGEM WIENER

Monografia sob a orientação do Prof. MC. Claudio Penedo de Albuquerque, avaliada por uma Banca Examinadora do Curso de Engenharia da Computação da Faculdade de Ciências Exatas e Tecnologia - FAET do Centro Universitário de Brasília - UniCEUB e constituiu requisito para obtenção do Título de Bacharel em Engenharia da Computação.

Brasília, DF - fevereiro de 2005.

**“Mais valem as lágrimas de não ter vencido,
do que a vergonha de não ter lutado”.**

Agradecimentos

Aos **meus amados pais**, Fernando Francisco Alves e Angela Maria Kinuko Sakamoto Alves, por estarem sempre ao meu lado ensinando o verdadeiro valor do estudo e da dedicação, por serem pessoas tão especiais, pelo amor incondicional e por sempre acreditarem em mim.

Ao **meu irmão**, Jean Sakamoto Alves, por ter me apresentado a informática e por estar sempre me ensinando e me ajudando em todos os momentos.

Ao **meu namorado**, Bernardo Luiz Braga Coelho Júnior, por estar ao meu lado durante os últimos quatro anos, por compreender alguns momentos de ausência, pela paciência, por sempre ter respostas às minhas dúvidas e pelo incentivo constante.

Aos **colegas de faculdade**, por me ajudarem, ensinarem, confiarem, acompanharem e por fazerem estes cinco anos de curso mais divertidos. Agradeço principalmente aos amigos Adriano Delfino de Medeiros e Marcos Thompson Viegas Lerário por toda a ajuda, a colaboração, o apoio e a amizade a mim dispensada.

À **amiga** e grande engenheira Soraya Chaibub Araújo de Lemos por todo o carinho e atenção com que sempre me trata.

Aos **professores**, por nos transmitirem seus conhecimentos com tanta sapiência e dedicação. Agradeço ao Professor Francisco Javier de Obaldía por toda consideração, exclusividade e atenção com que trata seus alunos. Agradeço o Professor MC. Claudio Penedo de Albuquerque pela orientação na realização do presente projeto.

E agradeço a **Deus** por permitir que existam pessoas tão especiais em minha vida.

Resumo

Este estudo trata da restauração de imagens digitais. Serão apresentadas, discutidas e comparadas duas técnicas fundamentais de restauração: filtragem inversa e filtragem Wiener. O objetivo deste trabalho é abordar a modelagem matemática e simular, com o uso da ferramenta Matlab, essas duas técnicas conhecidas de restauração de imagens digitais, com a finalidade de apresentar as vantagens e desvantagens de cada uma, considerando-se imagens de natureza bidimensional discreta e monocromática.

Tendo como ponto principal a análise de tais técnicas, realiza-se uma introdução sobre as características da visão humana, sobre imagens digitais e sobre seu processo de digitalização. Em seguida, o estudo é ordenado no sentido de apresentar as duas técnicas citadas no parágrafo anterior, sendo tal apresentação baseada em suas formulações matemáticas, seguida pela apresentação de resultados obtidos e os modelos de validação em Matlab. Por fim, comparam-se as duas técnicas estudadas, a fim de expor as vantagens e desvantagens, bem como o desempenho de cada uma das técnicas.

Palavras-Chaves: Processamento digital de imagens, restauração, filtragem Wiener, filtragem inversa.

Abstract

This work is about digital image restoration. Two fundamental techniques of restoration will be presented, argued and compared: inverse filter and Wiener filter. The intended objective was to approach mathematically and to simulate this two known techniques of digital image restoration, with the purpose of to show the differences of each one, in the case of images with two-dimensional, discret and monochromatic nature.

The principal point was the analyze of the cited techniques, and was realized a introduction about the characteristics of the human vision, about digital images and the digitalization process. After that, the study was commanded in the way to present the mathematical formulation of the two techniques above cited, where this presentation was based in mathematical formulations, followed by the sample of gotten results and the models of Matlab's validation. Finally, there was a comparison between the studied techniques, in order to display the differences and performance of each one.

Key-Words: Digital image processing, image restoration, Wiener filter, inverse filter.

Sumário

Lista de Símbolos	X
Lista de Siglas	XI
Lista de Tabelas	XII
Lista de Figuras	XIII
Lista de Figuras	XIII
Capítulo 1 – Introdução	1
1.1 – Motivação	1
1.2 – Objetivo do trabalho proposto	2
1.3 – Imagem utilizada	3
1.4 – Descrição dos capítulos	3
Capítulo 2 – A imagem digital	5
2.1 – Percepção visual humana.....	5
2.2 – Conceito de imagem digital.....	6
2.3 – Representação da Imagem Digital.....	9
2.4 – Domínio da Frequência	10
2.4.1 – Transformada Discreta de Fourier Bidimensional (DFT)	11
2.4.2 – Transformada Rápida de Fourier Bidimensional (DFFT).....	12
2.4.3 – Resposta à Frequência	14
2.5 –Imagem utilizada no projeto	16
2.5.1 – Características da imagem.....	17
Capítulo 3 – Restauração de imagens	19
3.1 –A Restauração.....	19
3.1.1 – A técnica da Restauração.....	19
3.2 – Modelo de Degradação	20
3.2.1 – Embaçamento (“ <i>Blurring</i> ”)	21
3.2.2 – Ruído aditivo	23
3.2.3 – Definição de $h[x,y]$	24
3.3 – Modelo de Restauração	25
Capítulo 4 – Filtragem no domínio da frequência.....	26
4.1 – Conceito.....	26
4.2 - Filtragem inversa.....	27

4.2.1 – Filtragem inversa – procedimento prático	27
4.2.2 – Vantagens e desvantagens	28
4.3 - Filtragem Wiener.....	29
4.3.1 – Filtragem Wiener – procedimento prático	32
4.3.2 – Vantagens e Desvantagens.	33
4.4 Motivos que levaram a escolha dos métodos de restauração	34
Capítulo 5 – Simulação	35
5.1 – Amostra	35
5.2 – Configuração do ambiente de homologação	35
5.3 – Métodos de análise	36
5.3.1 – Histogramas	36
5.3.2 – SNR e ISNR	37
5.4 – Descrição do código fonte por diagramas	38
5.4.1 – Inserção de embaçamento na imagem	38
5.4.2 – Inserção de ruído na imagem	39
5.4.3 – Filtragem inversa	40
5.4.4 – Filtragem Wiener	41
Capítulo 6 – Análise de Resultados	44
6.1 - Resultados – Embaçamento e Ruído	44
6.2 - Resultados – Filtragem inversa	46
6.2.1 - Média.....	48
6.2.2 – Desvio Padrão	49
6.2.3 - ISNR.....	49
6.2.4 – Análise Geral	50
6.3 - Resultados – Filtragem Wiener	51
6.3.1 – Média, Desvio Padrão e ISNR.....	51
6.3.2 – Análise Geral	52
6.4 – Resultados - Conclusão	53
6.5 – Melhores resultados obtidos	54
6.5.1 – Filtragem inversa	54
6.5.2 – Filtragem Wiener	55
Capítulo 7 - Conclusão	56
7.1 – Considerações finais	56
7.2 – Trabalhos Futuros	57

Referências Bibliográficas	58
ANEXO A – Código Fonte em Matlab	62
ANEXO B – Imagem original e degradadas	74
ANEXO C – Filtragem inversa	79
ANEXO D – Filtragem Wiener	97

Lista de Símbolos

$*$: Operador linear de convolução.

Σ : Somatório.

$()$: Representação de espaço contínuo.

$[]$: Representação de espaço discreto.

(x,y) : Representação no domínio do espaço contínuo.

$[x,y]$: Representação no domínio do espaço discreto.

$[u,v]$: Representação no domínio da frequência.

$F[x,y]$: Imagem original no domínio do espaço.

$F[u,v]$: Imagem original no domínio da frequência.

$G[x,y]$: Imagem degradada no domínio do espaço.

$G[u,v]$: Imagem degradada no domínio da frequência.

$h[x,y]$: Função de espalhamento de ponto (PSF) no domínio do espaço.

$H[u,v]$: Função de espalhamento de ponto (PSF) no domínio da frequência.

$\eta[x,y]$: Ruído no domínio do espaço.

$N[u,v]$: Ruído no domínio da frequência.

$M \times N$: Matriz com M linhas e N colunas.

Lista de Siglas

DFT: Transformada Discreta de Fourier.

FFT: Transformada Rápida de Fourier.

FFTW (“*Fastest Fourier Transform in the West*”): Mais Rápida Transformada de Fourier no Ocidente.

FT: Transformada de Fourier.

IEEE (“*Institute of Electrical and Electronics Engineers*”): Instituto de Engenheiros Eletricistas e Eletrônicos.

ISNR (“*Improvement in Signal to Noise Ratio*”): Melhoria na Relação Sinal-Ruído.

JPEG (“*Joint Photographic Experts Group*”)

PDI: Processamento Digital de Imagens.

PSF (“*Point Spread Function*”): Função de Espalhamento de Ponto.

Pixel (“*Picture Element*”): Elemento de Imagem.

QoS (“*Quality of Service*”): Qualidade de Serviço.

RMS (“*Root Mean Square*”): Raiz Quadrada Média.

UniCEUB: Centro Universitário de Brasília.

Lista de Tabelas

Tabela 2.1 – Características da imagem utilizada no trabalho.	18
Tabela 6.1 – Resultados: imagens com degradação.	44
Tabela 6.2 – Resultados: filtragem inversa.	47
Tabela 6.3 – Resultados: filtragem Wiener.	51

Lista de Figuras

Figura 1.1 – Lenna.jpg - Imagem utilizada no trabalho.	3
Figura 2.1 – Espectro Eletromagnético e range de luz visível ao olho humano.....	5
Figura 2.2 – Eixos x e y de uma imagem digitalizada.....	7
Figura 2.3 – (a) Sensor de imagem; (b) Sensores em linha; (c) Sensores em array.	8
Figura 2.4 – Digitalização da imagem.....	8
Figura 2.5 - 256 níveis de cinza.....	9
Figura 2.6 – Imagem digital e sua matriz.	9
Figura 2.7 – Prisma.....	10
Figura 2.8 – FFT da Figura 1.1.....	13
Figura 2.9 – Decomposição da FFT.	14
Figura 2.10 – Comparação entre sistemas no domínio espacial e da frequência.	16
Figura 3.1 – Modelo do processo de degradação e restauração de uma imagem.....	20
Figura 3.2 – Modelo de degradação.	21
Figura 3.3 – Imagem original embaçada.	23
Figura 3.4 – Modelo de restauração.	25
Figura 4.1 – Passos básicos da filtragem no domínio da frequência.....	26
Figura 4.2 – Filtro inverso.	27
Figura 5.1 – Imagem Original.	35
Figura 5.2 – Histograma da imagem original.....	36
Figura 5.3 – Diagrama – Inserção de embaçamento na imagem.....	39
Figura 5.4 –Inserção de ruído na imagem.	40
Figura 5.5 – Diagrama – Filtragem inversa.....	41
Figura 5.6 – Diagrama – Filtragem Wiener.....	43
Figura 6.1 – Gráfico – Imagens com degradação – Média.....	45
Figura 6.2 – Gráfico – Imagens com degradação – Desvio Padrão.	46
Figura 6.3 – Gráfico – Filtragem inversa – Média.	48
Figura 6.4 – Gráfico – Filtragem inversa – Desvio Padrão.....	49
Figura 6.5 – Gráfico – Filtragem inversa – ISNR.	50
Figura 6.6 – Gráfico – Filtragem Wiener – Média, Desvio Padrão e ISNR.....	52
Figura 6.7 – Gráfico - Filtragem inversa x Wiener – melhores resultados.	53
Figura 6.8 – Filtragem Wiener – melhor resultado.	55

Figura B.1 – Imagem original e seu histograma.....	74
Figura B.2 – Imagem original embaçada e seu histograma.....	74
Figura B.3 – Imagem degradada com SNR = 5 dB e seu histograma.	75
Figura B.4 – Imagem degradada com SNR = 10 dB e seu histograma.	75
Figura B.5 – Imagem degradada com SNR = 20 dB e seu histograma.	75
Figura B.6 – Imagem degradada com SNR = 30 dB e seu histograma.	76
Figura B.7 – Imagem degradada com SNR = 40 dB e seu histograma.	76
Figura B.8 – Imagem degradada com SNR = 50 dB e seu histograma.	76
Figura B.9 – Imagem degradada com SNR = 60 dB e seu histograma.	77
Figura B.10 – Imagem degradada com SNR = 70 dB e seu histograma.	77
Figura B.11 – Imagem degradada com SNR = 80 dB e seu histograma.	77
Figura B.12 – Imagem degradada com SNR = 90 dB e seu histograma.	78
Figura B.13 – Imagem degradada com SNR = 100 dB e seu histograma.	78
Figura C.1 – Filt. inversa, Imagem Original, SNR=0dB, thresh=0.000001, ISNR= 228.35dB e seu histograma.	79
Figura C.2 – Filt. inversa, Imagem Original, SNR=0dB, thresh=0.05, ISNR= 5.06dB e seu histograma.	79
Figura C.3 – Filt. inversa, Imagem Original, SNR=0dB, thresh=0.1, ISNR= 3.38dB e seu histograma.	80
Figura C.4 – Filt. inversa, Embaçam.+ SNR=5dB, thresh=0.000001, ISNR= -76.66dB e seu histograma.	80
Figura C.5 – Filt. inversa, Embaçam.+ SNR=5dB, thresh=0.05, ISNR= -11.06dB e seu histograma.	81
Figura C.6 – Filt. inversa, Embaçam.+ SNR=5dB, thresh=0.1, ISNR= -2.57dB e seu histograma.	81
Figura C.7 – Filt. inversa, Embaçam.+ SNR=10dB, thresh=0.000001, ISNR= -73.42dB e seu histograma.	82
Figura C.8 – Filt. inversa, Embaçam.+ SNR=10dB, thresh=0.05, ISNR= -8.22dB e seu histograma.	82
Figura C.9 – Filt. inversa, Embaçam.+ SNR=10dB, thresh=0.1, ISNR= -0.45dB e seu histograma.	83
Figura C.10 – Filt. inversa, Embaçam.+ SNR=20dB, thresh=0.000001, ISNR= -63.22dB e seu histograma.	83

Figura C.11 – Filt. inversa, Embaçam.+ SNR=20dB, thresh=0.05, ISNR= -0.84dB e seu histograma.	84
Figura C.12 – Filt. inversa, Embaçam.+ SNR=20dB, thresh=0.1, ISNR= 2.59dB e seu histograma.	84
Figura C.13 – Filt. inversa, Embaçam.+ SNR=30dB, thresh=0.000001, ISNR= -53.67dB e seu histograma.	85
Figura C.14 – Filt. inversa, Embaçam.+ SNR=30dB, thresh=0.05, ISNR= 3.94dB e seu histograma.	85
Figura C.15 – Filt. inversa, Embaçam.+ SNR=30dB, thresh=0.1, ISNR= 3.29dB e seu histograma.	86
Figura C.16 – Filt. inversa, Embaçam.+ SNR=40dB, thresh=0.000001, ISNR= -44.15dB e seu histograma.	86
Figura C.17 – Filt. inversa, Embaçam.+ SNR=40dB, thresh=0.05, ISNR= 4.93dB e seu histograma.	87
Figura C.18 – Filt. inversa, Embaçam.+ SNR=40dB, thresh=0.1, ISNR= 3.37dB e seu histograma.	87
Figura C.19 – Filt. inversa, Embaçam.+ SNR=50dB, thresh=0.000001, ISNR= -31.12dB e seu histograma.	88
Figura C.20 – Filt. inversa, Embaçam.+ SNR=50dB, thresh=0.05, ISNR= 5.05dB e seu histograma.	88
Figura C.21 – Filt. inversa, Embaçam.+ SNR=50dB, thresh=0.1, ISNR= 3.37dB e seu histograma.	89
Figura C.22 – Filt. inversa, Embaçam.+ SNR=60dB, thresh=0.000001, ISNR= -24.34dB e seu histograma.	89
Figura C.23 – Filt. inversa, Embaçam.+ SNR=60dB, thresh=0.05, ISNR= 5.06dB e seu histograma.	90
Figura C.24 – Filt. inversa, Embaçam.+ SNR=60dB, thresh=0.1, ISNR= 3.38dB e seu histograma.	90
Figura C.25 – Filt. inversa, Embaçam.+ SNR=70dB, thresh=0.000001, ISNR= -10.45dB e seu histograma.	91
Figura C.26 – Filt. inversa, Embaçam.+ SNR=70dB, thresh=0.05, ISNR= 5.06dB e seu histograma.	91
Figura C.27 – Filt. inversa, Embaçam.+ SNR=70dB, thresh=0.1, ISNR= 3.38dB e seu histograma.	92

Figura C.28 – Filt. inversa, Embaçam.+ SNR=80dB, thresh=0.000001, ISNR= -0.42dB e seu histograma.	92
Figura C.29 – Filt. inversa, Embaçam.+ SNR=80dB, thresh=0.05, ISNR= 5.06dB e seu histograma.	93
Figura C.30 – Filt. inversa, Embaçam.+ SNR=80dB, thresh=0.1, ISNR= 3.38dB e seu histograma.	93
Figura C.31 – Filt. inversa, Embaçam.+ SNR=90dB, thresh=0.000001, ISNR= 7.29dB e seu histograma.	94
Figura C.32 – Filt. inversa, Embaçam.+ SNR=90dB, thresh=0.05, ISNR= 5.06dB e seu histograma.	94
Figura C.33 – Filt. inversa, Embaçam.+ SNR=90dB, thresh=0.1, ISNR= 3.38dB e seu histograma.	95
Figura C.34 – Filt. inversa, Embaçam.+ SNR=100dB, thresh=0.000001, ISNR= 17.82dB e seu histograma.	95
Figura C.35 – Filt. inversa, Embaçam.+ SNR=100dB, thresh=0.05, ISNR= 5.06dB e seu histograma.	96
Figura C.36 – Filt. inversa, Embaçam.+ SNR=100dB, thresh=0.1, ISNR= 3.38dB.....	96
Figura D.1 – Filt. Wiener, Imagem Original, ISNR= 228.35dB e seu histograma.	97
Figura D.2 – Filt. Wiener, Embaçam.+SNR=5dB, ISNR= 5.07dB e seu histograma.....	97
Figura D.3– Filt. Wiener, Embaçam.+SNR=10dB, ISNR= 3.83dB e seu histograma.....	98
Figura D.4– Filt. Wiener, Embaçam.+SNR=20dB, ISNR= 4.14dB e seu histograma.....	98
Figura D.5 – Filt. Wiener, Embaçam.+SNR=30dB, ISNR= 6.04dB e seu histograma.....	99
Figura D.6 – Filt. Wiener, Embaçam.+SNR=40dB, ISNR= 8.42dB e seu histograma.....	99
Figura D.7 – Filt. Wiener, Embaçam.+SNR=50dB, ISNR= 11.38dB e seu histograma.....	100
Figura D.8 – Filt. Wiener, Embaçam.+SNR=60dB, ISNR= 14.72dB e seu histograma.....	100
Figura D.9 – Filt. Wiener, Embaçam.+SNR=70dB, ISNR= 18.47dB e seu histograma.....	101
Figura D.10 – Filt. Wiener, Embaçam.+SNR=80dB, ISNR= 22.93dB e seu histograma.....	101
Figura D.11 – Filt. Wiener, Embaçam.+SNR=90dB, ISNR= 27.72dB e seu histograma.....	102
Figura D.12 – Filt. Wiener, Embaçam.+SNR=100dB, ISNR= 32.98dB e seu histograma....	102

Capítulo 1 – Introdução

1.1 – Motivação

A visão é um poderoso atributo de percepção. Uma das partes fundamentais da sensação visual são as imagens.

A percepção visual humana é baseada no processamento da imagem percebida pelo cérebro em associação com informações já armazenadas e um raciocínio dedutivo. Logo, o cérebro trabalha interpretando as sensações visuais com base no que já está em uma espécie de acervo cerebral, como tamanho, forma, cor e comportamento. Portanto, o fato de que a visualização de uma imagem danificada pode ser compreendida, é real.

Diversas áreas fazem uso do Processamento Digital de Imagens (PDI), dentre elas:

- A engenharia, atuando na mais diversas áreas científicas e tecnológicas que necessitam utilizar imagens;
- A medicina, com a ressonância magnética, ecocardiografia, endoscopia, etc.;
- A arqueologia, trabalhando com documentos e imagens antigas;
- Os estudos físicos destacando-se os estudos matemáticos e os biológicos;
- As produções industriais, com imagens geradas por sensores visuais em robôs;
- O geoprocessamento que abrange áreas como a cartografia, a geologia, a geografia, as imagens de satélites, etc.;
- As imagens de perícias criminais, dentre outras.

Entre 1950 e 1960, a restauração de imagens teve sua utilização iniciada com programas espaciais dos Estados Unidos e da antiga União Soviética [WEN, 2002]. Desse momento em diante, a tecnologia de restauração foi naturalmente estendida a outras áreas.

Com esse impulso por melhorias nas imagens, o avanço da tecnologia com técnicas mais elaboradas e equipamentos mais robustos, foi proporcionado um grande benefício para as técnicas de restauração. Tornou-se constante a busca por melhorias nas técnicas de PDI e conseqüentemente de uma melhor apresentação visual das imagens.

A restauração de imagens é uma área que visa melhorar a aparência de uma imagem degradada. É uma técnica objetiva que parte do princípio de que uma imagem com boa

visualização é aquela em que os efeitos degradantes podem ser corrigidos a ponto de se obter uma imagem com defeitos mínimos ou quase imperceptíveis ao olho humano.

Tais fenômenos de degradação podem advir de embaçamentos e ruídos eletrônicos [WEN, 2002], que muitas vezes são inevitáveis. Mas a melhoria na aquisição de tais imagens pode se tornar cara ou não muito conveniente como, por exemplo, para se obter melhor qualidade nas imagens de raios-x é necessário que se aumente a intensidade do feixe de radiação, o que é perigoso à saúde da pessoa submetida a tal processo, ou ainda, o aumento da capacidade de processamento dos computadores utilizados em tal processo. No atual cenário mundial é interessante que se desenvolvam algoritmos¹ que não necessitem de grandes custos de processamento, tornando as técnicas de restauração menos complexas.

Apesar de muitas vezes tal prática se tornar cara e demorada, a restauração de imagens é uma solução confiável de obtenção de uma melhor aproximação da imagem real, utilizando várias técnicas dentre as quais são enfocadas duas no presente estudo: filtragem inversa e filtragem Wiener.

1.2 – Objetivo do trabalho proposto

A ligação entre a engenharia da computação e as demais áreas citadas na seção 1.1, justifica-se pelo fato de que a computação está presente atualmente em todas as áreas do conhecimento. Tais áreas utilizam e dependem do PDI.

A restauração de imagens, portanto, é uma subárea intrínseca e necessária ao PDI.

O trabalho proposto tem por objetivo:

- Modelar matematicamente alguns conceitos básicos para a técnica de restauração;
- Modelar matematicamente os modelos de degradação e restauração de imagens;
- Modelar matematicamente as degradações aplicadas: ruído e embaçamento;
- Desenvolver um algoritmo que faça a inserção da degradação em uma imagem original;
- Desenvolver algoritmos para o filtro inverso e filtro Wiener, seguindo a modelagem matemática apresentada;

¹ Algoritmo: conjunto das regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema em um número finito de etapas. Fonte: [45].

- Simular utilizando o software Matlab² a restauração da imagem, aplicando a degradação e os filtros de restauração.
- Comparar os resultados obtidos entre os dois filtros utilizados e realizar as devidas conclusões.
- Contribuir com o auxílio a pesquisas acadêmicas e servir como base para futuros projetos na área da Engenharia da Computação.

1.3 – Imagem utilizada

A imagem que será utilizada nas simulações deste trabalho é apresentada na Figura 1.1.



Figura 1.1 – Lenna.jpg - Imagem utilizada no trabalho.

Fonte: [KOKARAM, 2003].

1.4 – Descrição dos capítulos

Este trabalho está dividido em sete capítulos:

- **Capítulo 1:** Faz uma apresentação introdutória sobre o tema restauração de imagens e a motivação sobre o desenvolvimento do presente trabalho, descreve o objetivo do trabalho proposto e apresenta a imagem a ser utilizada nas simulações.

² Matlab: software proprietário da *MathWorks Inc.*, que utiliza uma linguagem computacional de alto nível e de desenvolvimento interativo para o desenvolvimento de algoritmos, visualização de dados, análise de dados e computação numérica. Fonte: [44].

- **Capítulo 2:** Aborda a imagem digital e suas características, além de apresentar propriedades da percepção visual humana, fenômenos físicos da formação de imagens e conceitos básicos sobre imagens. Além disso, existe o conceito da Transformada Discreta de Fourier Bidimensional que é a base principal dos filtros utilizados.
- **Capítulo 3:** Neste capítulo, encontra-se o tema principal deste projeto: a restauração de imagens. São apresentados os modelos de degradação e de restauração da imagem, os tipos de degradação utilizados – embaçamento e ruído – e suas modelagens matemáticas.
- **Capítulo 4:** É apresentada a filtragem digital, com o detalhamento dos filtros inverso e Wiener. Ainda neste capítulo são abordados os motivos que levaram à escolha de tais filtros.
- **Capítulo 5:** Descreve a simulação realizada em Matlab. Neste capítulo são apresentadas as descrições do código fonte em diagramas de fluxo³, a configuração do ambiente de homologação e quais os motivos da escolha do software Matlab para as simulações.
- **Capítulo 6:** Contém as análises de desempenho das simulações, dentre elas: os histogramas⁴ de níveis de cinza, a relação sinal-ruído, o desempenho obtido pelos filtros para o tipo de degradação utilizado, a aplicação dos filtros com suas vantagens e desvantagens, e ainda, uma tabela comparativa entre os métodos.
- **Capítulo 7:** Expõe as considerações finais do trabalho e destaca idéias para projetos futuros.

³ Diagrama de fluxo: diagrama para representação de um algoritmo; fluxograma. Fonte: [45]

⁴ Histograma: distribuição dos valores dos *pixels* de acordo com suas incidências na imagem.

Capítulo 2 – A imagem digital

2.1 – Percepção visual humana

Em 1666, Isaac Newton descobriu que a passagem de um feixe de luz do Sol através de um prisma resultava em um espectro de luz contínuo variando do vermelho ao violeta [GONZALEZ, 2002]. A partir de então, o conceito de espectro eletromagnético aborda uma faixa de frequências, onde a parte visível se encontra entre as frequências de $4,3 \times 10^{14}$ Hz a 7×10^{14} Hz. Como pode ser visto na Figura 2.1, a faixa de cores perceptíveis na luz visível representa uma faixa muito pequena do espectro eletromagnético [GONZALEZ, 2002].

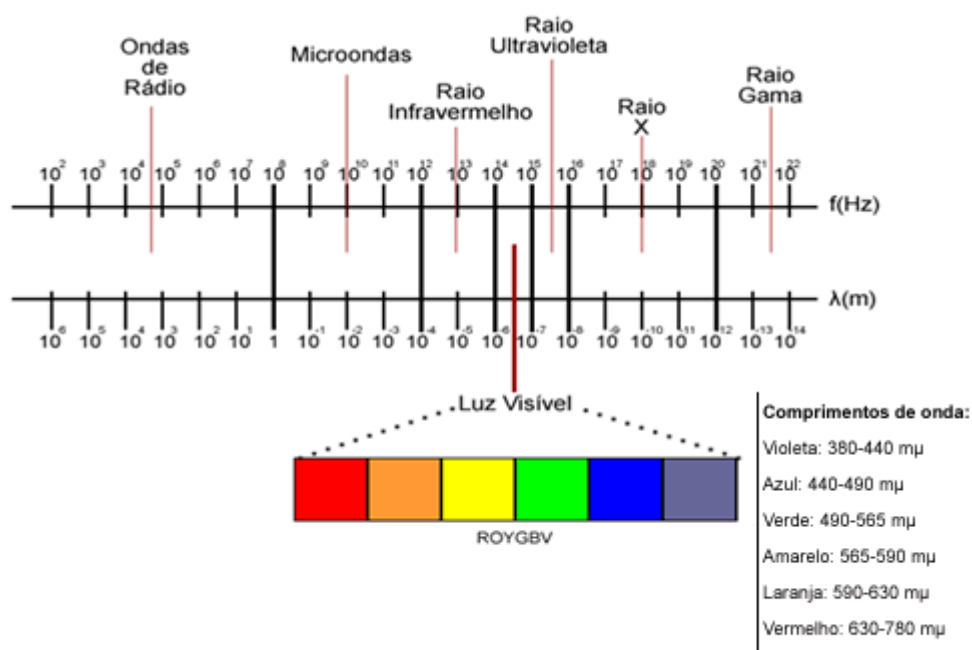


Figura 2.1 – Espectro Eletromagnético e range de luz visível ao olho humano.

Para compreender como uma imagem é percebida visualmente pelo olho humano, é interessante fazer uma análise física da luz. A luz é entendida como uma energia eletromagnética que incide sobre os corpos no espaço. A partir da reflexão dessa luz incidente, os corpos podem ser vistos apresentando a cor correspondente à frequência da luz refletida.

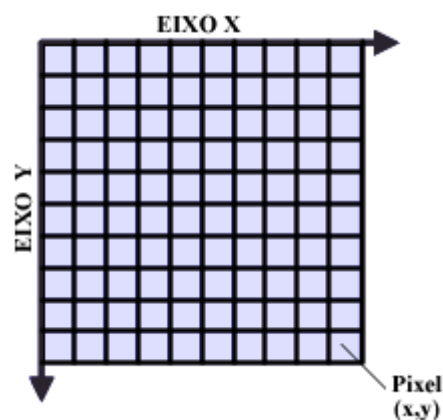
2.2 – Conceito de imagem digital

Uma imagem pode então ser entendida como um reflexo ou radiação emitida de um objeto ou de uma cena. Esta imagem é armazenada em um computador como um conjunto de pontos de cor denominados *pixels*⁵, cada um representado por um número. Assim, um conjunto de *pixels* forma uma matriz de números obtida por amostragem e quantização dos dados em forma contínua, inicialmente adquiridos.

A amostragem é a digitalização⁶ das posições discretas das coordenadas de cada *pixel*. Essas coordenadas são relacionadas com um sistema de referência (plano cartesiano, projeção cartográfica). Já a quantização é a digitalização dos valores de suas amplitudes [GONZALEZ, 2002].

Matematicamente, a imagem digital é uma função em duas dimensões, dada por $f[x, y]$ em que x e y são as coordenadas espaciais. O valor de $f[x, y]$ em qualquer ponto $[x, y]$ é proporcional ao nível de cinza da imagem naquele ponto [CARVALHO, 2004]. Logo, para uma matriz de dimensão finita formada pelos valores de correspondentes aos da imagem digitalizada, tem-se então, uma imagem digital.

A Figura 2.2 mostra a matriz formada por cada *pixel* da imagem, com suas coordenadas x e y em um plano cartesiano. Como uma imagem é formada por finitos números de *pixels*, logo, imagens grandes podem causar um problema computacional em seu processamento, por necessitarem de grande disponibilidade de memória, espaço de armazenamento e equipamentos adequados.



⁵ Pixel (“*Picture Element*”): é o termo mais utilizado para denotar cada um dos finitos elementos de uma imagem digital. Cada pixel possui um valor e uma localização particular. Fonte: [1].

⁶ Digitalização: conversão de um sinal analógico em um sinal digital [54].

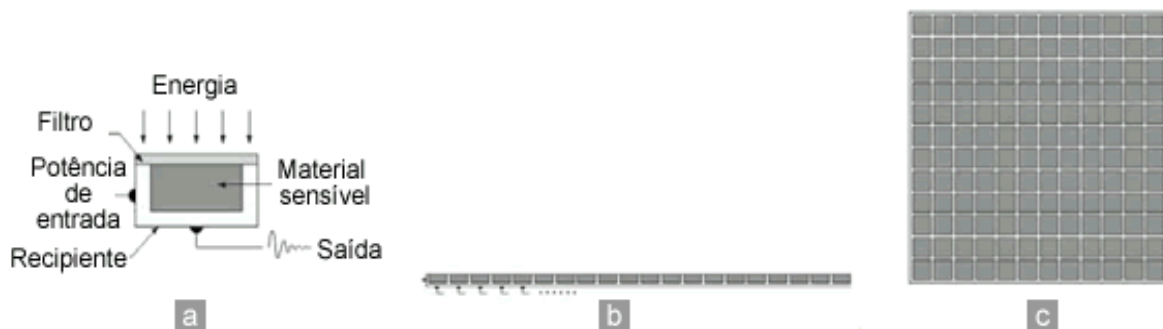
Figura 2.2 – Eixos x e y de uma imagem digitalizada.

O processo de aquisição de uma imagem é feito inicialmente através de sensores instalados nos equipamentos utilizados. A Figura 2.3 mostra as três formas de organização de sensores mais utilizadas para a transformação de energia luminosa em imagens digitais. Cada uma dessas formas tem a finalidade de aplicar a técnica de amostragem e quantização da imagem, onde cada ponto da imagem é definido como um *pixel*.

Na Figura 2.3 (a), têm-se representado um único sensor. O uso do filtro na parte anterior ao material sensível produz uma maior seletividade da entrada (do tipo de forma de onda determinado para o filtro). A produção de uma imagem em duas dimensões por um único sensor é uma prática bastante demorada, mas que produz resultados em alta definição [GONZALEZ, 2002]. Tal afirmação se explica pelo fato de que, como o sensor captura cada ponto da imagem, logo, a imagem não tem uma grande perda de dados no momento de sua digitalização. O sensor unitário pode ser utilizado em conjunto formado alternativas de captação de imagem, como mostrado nas figuras 2.3 (b) e (c).

A Figura 2.3 (b) mostra um arranjo de sensores em linha, que é bastante utilizado na aquisição de imagens por *scanners* de mesa (geralmente utilizam 4000 ou mais sensores em linha) [GONZALEZ, 2002]. Essa disposição de sensores é ainda utilizada em forma circular, como em aparelhos de tomografia computadorizada utilizando-se raios-X [GONZALEZ, 2002].

Na Figura 2.3 (c) está disposta a combinação de sensores em forma de um *array*⁷ bidimensional. Essa é a forma predominantemente utilizada em câmeras digitais. A dimensão desse *array* é geralmente de 4000x4000 sensores, ou mais. A resposta de cada sensor é proporcional à integral da energia luminosa projetada na superfície do sensor. Já que nesse caso a disposição dos sensores é bidimensional, a imagem completa pode ser obtida focando a energia incidente sobre a superfície do *array*.



⁷ Array: É uma sequência de elementos.

Figura 2.3 – (a) Sensor de imagem; (b) Sensores em linha; (c) Sensores em array.

Fonte: [GONZALEZ, 2002], com adaptações.

A Figura 2.4 mostra a principal maneira de captação de imagem por sensores dispostos em *array*. A energia de uma fonte de iluminação é refletida em um elemento da cena. É importante ressaltar que o próprio elemento da cena pode transmitir sua própria luminosidade (por exemplo, uma luminária).

O sistema de captação da imagem coleta a energia incidente e a enfoca sobre um plano da imagem. Os sensores em *array*, que coincidem com os pontos do plano da imagem, produzem saídas proporcionais à integral da luz recebida por cada sensor. Logo, a saída desse sistema é representada por uma imagem digitalizada [GONZALEZ, 2002].

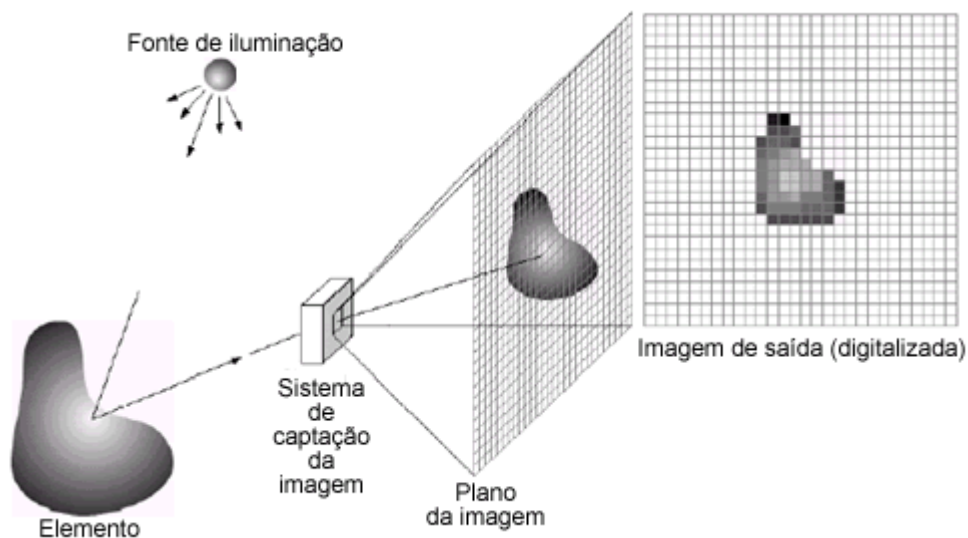


Figura 2.4 – Digitalização da imagem.

Fonte: [GONZALEZ, 2002], com adaptações.

Os tons de cor da imagem final digitalizada são dados pela técnica de quantização. De acordo com a amplitude da frequência dos valores amostrados para cada *pixel* da imagem original, uma tabela de tons de cor em níveis de cinza é utilizada para uma comparação de valores que serão aplicados na definição dos valores dos *pixels* da imagem digitalizada. Ou seja, os valores amostrados aproximados aos da tabela escolhida definirão, levando em conta possíveis erros de arredondamento, os tons de cor de cada *pixel*.

A Figura 2.5 demonstra uma escala de 256 níveis de cinza, onde o valor 0 (zero) representa a cor preta e o valor 255 representa a cor branca. Entre estes valores há a variação na tonalidade de cinza, como pode ser observado.



Figura 2.5 - 256 níveis de cinza.

2.3 – Representação da Imagem Digital

O estudo da imagem digital depende de sua associação com um modelo matemático e sua representação discreta inteligível ao computador.

De maneira simplificada, o processo é descrito como tendo a princípio uma imagem adquirida representada como uma função bidimensional discreta. Em seguida, após sua discretização adquire-se uma imagem bidimensional em domínio discreto e a codificação de tal imagem é dada por uma matriz de números, como pode ser observada na Figura 2.6, onde cada elemento da matriz corresponde ao valor do nível de cinza do *pixel* correspondente na imagem.

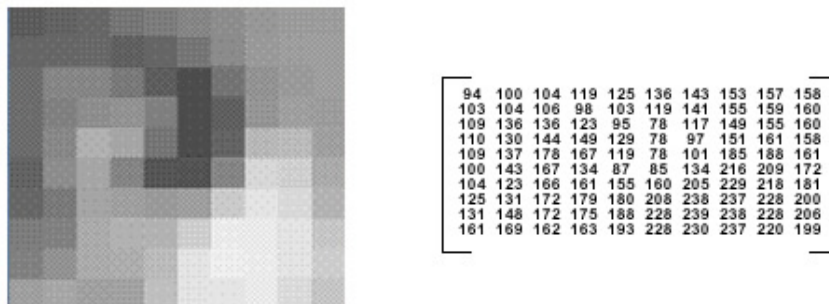


Figura 2.6 – Imagem digital e sua matriz.

Uma imagem monocromática $f[x,y]$ resulta em uma imagem digital de M linhas por N colunas. Os valores de $[x,y]$ são dados em quantidades discretas.

Uma imagem digital é representada numericamente por:

$$f[x,y] = \begin{bmatrix} f[0,0] & f[0,1] & \cdots & f[0,N-1] \\ f[1,0] & f[1,1] & \cdots & f[1,N-1] \\ \vdots & \vdots & \cdots & \vdots \\ f[M-1,0] & f[M-1,1] & \cdots & f[M-1,N-1] \end{bmatrix}_{M \times N} \quad (2.1)$$

O lado direito da Equação (2.1) – representação matricial - é, por definição, a representação da estrutura matemática de uma imagem digital [GONZALEZ, 2002].

2.4 – Domínio da Frequência

Jean Baptiste Fourier (1768 - 1830) fez a descoberta de que uma função que se repete periodicamente pode ser expressa como uma soma de senos e/ou cossenos de diferentes frequências [GONZALEZ, 2002]. Esse somatório é conhecido como Série de Fourier. Tal soma de termos não leva em consideração o tamanho da função, ou seja, desde que a função seja periódica a mesma pode ser representada por tal soma.

Jean Baptiste Fourier apresentou ainda o fato de que mesmo funções que não são periódicas podem ser expressas como uma integral de senos e/ou cossenos. Esta afirmação é válida para os casos de funções finitas [GONZALEZ, 2002]. Para este caso, as formulações são conhecidas como Transformadas de Fourier (FT).

Uma analogia interessante é a comparação da FT com um prisma de vidro. Como pode ser observado na Figura 2.7, o prisma é uma ferramenta física que separa a luz em seus componentes de cor, de acordo com seus comprimentos de onda ou frequência [GONZALEZ, 2002].

A FT pode ser vista como um “prisma de frequência” que separa uma função em suas várias componentes, baseando-se em seu conteúdo de frequências.

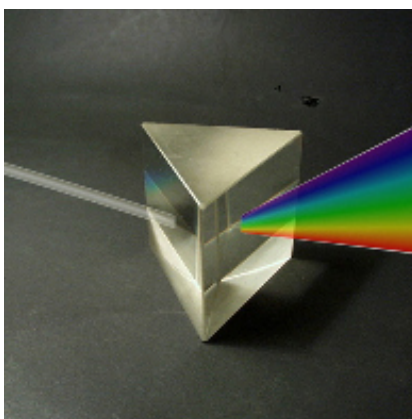


Figura 2.7 – Prisma.

Fonte: [BERROCOSO, 2003].

Um aspecto interessante ao se utilizar a FT em PDI, é o fato de que funções que sofreram sua aplicação podem ser perfeitamente reconstruídas, pelo processo inverso. Ou seja, uma função em domínio espacial após a aplicação da FT é analisada em domínio da frequência. Aplicando-se a FT inversa tem-se novamente a função em domínio espacial, sem perda significativa de suas informações. Esse é um dos principais motivos de se trabalhar no chamado domínio Fourier ou domínio da frequência.

Este trabalho será baseado na restauração de imagens no domínio da frequência, logo, terá como base a Transformada Discreta de Fourier Bidimensional (DFT) já que permite, portanto, trabalhar com as imagens no domínio da frequência. A DFT será explicada com mais detalhes no tópico 2.4.1.

A DFT serve como conceito base para a formulação matemática dos filtros escolhidos para utilização neste trabalho. Mas a aplicação prática é dada utilizando-se algoritmos com a Transformada Rápida de Fourier (FFT). A FFT foi desenvolvida por volta dos anos 50 e, a partir de então, revolucionou o campo de processamento de sinais [GONZALEZ, 2002]. A FFT será explicada com mais detalhes no tópico 2.4.2.

A notação utilizada, neste trabalho, representa as funções em domínio espacial discreto por um padrão com fonte minúscula e coordenadas dadas por x e y (Exemplo: $f[x,y]$). Já as funções já transformadas para o domínio da frequência, seguem um padrão com fonte maiúscula e coordenadas dadas por u e v ($F[u,v]$).

2.4.1 – Transformada Discreta de Fourier Bidimensional (DFT)

Para o caso bidimensional, a DFT de uma função $f[x,y]$ no domínio espacial, de dimensão $M \times N$, onde $M=N$ (matriz quadrada), é dada por:

$$F[u,v] = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f[x,y] e^{-j2\pi \left(\frac{ux+vy}{N} \right)}, \quad (2.2)$$

para $u, v = 0, 1, \dots, N-1$, onde $f[x,y]$ é a imagem original, $F[u,v]$ é a representação da imagem original no domínio da frequência, M é o número de linhas da matriz e N é o número de colunas [GONZALEZ, 2002].

A transformada inversa da Equação (2.2) é dada por:

$$f[x, y] = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F[u, v] e^{j2\pi \left(\frac{ux+vy}{N} \right)} \quad (2.3)$$

para $x, y = 0, 1, \dots, M-1$ [GONZALEZ, 2002].

O conceito abordado da divisão de uma função em uma soma de senos e/ou cossenos, parte da Fórmula de Euler, apresentada na Equação (2.4).

$$e^{j\theta} = \cos \theta + j \sin \theta. \quad (2.4)$$

A Equação (2.4) pode ser substituída nas Equações (2.2) e (2.3), caso prefira-se trabalhar com os senos e cossenos ao invés do termo exponencial.

2.4.2 – Transformada Rápida de Fourier Bidimensional (DFFT)

A aplicação utilizada neste trabalho da Transformada Rápida de Fourier Bidimensional (FFT), segue o padrão utilizado pelo software Matlab. Tal padrão utiliza algoritmos que têm como base uma biblioteca denominada FFTW (*“Fastest Fourier Transform in the West”*) [FRIGO, 2004]. A FFTW é uma biblioteca de rotinas em linguagem de programação C, já desenvolvidas para o cálculo da DFT.

A FFTW não utiliza um algoritmo fixo para computar a DFT, mas adapta o algoritmo da DFT para os detalhes do hardware utilizado para maximizar sua performance [FRIGO, 2004].

Com isso, a computação da transformada é dividida em duas fases. Na primeira, é feito um plano de reconhecimento da maneira mais rápida de computar a transformada na máquina utilizada. É criado então uma estrutura de dados chamada “plano” que contém essa informação. Em seguida, o “plano” é executado com os dados de entrada para a realização da transformada [FRIGO, 2004].

Percebe-se, portanto, que a utilização da FFTW otimiza o cálculo de DFTs promovendo uma maior rapidez nos resultados. As rotinas em C da FFT utilizadas pela FFTW, não serão apresentadas neste trabalho por não serem o escopo principal do projeto.

Aplicando-se a FFT bidimensional na Figura 1.1, tem-se o resultado obtido apresentado na Figura 2.8. A figura mostra a distribuição de frequências no espectro da imagem utilizada. Observa-se que próximo ao centro da figura, no local onde se encontra a

origem dos pontos, existe a concentração de frequências mais altas representadas pela coloração mais clara. Por conveniência, utiliza-se a apresentação das frequências mais altas em torno da origem e sobre as divisões dos quadrantes, para uma melhor compreensão visual do espectro.

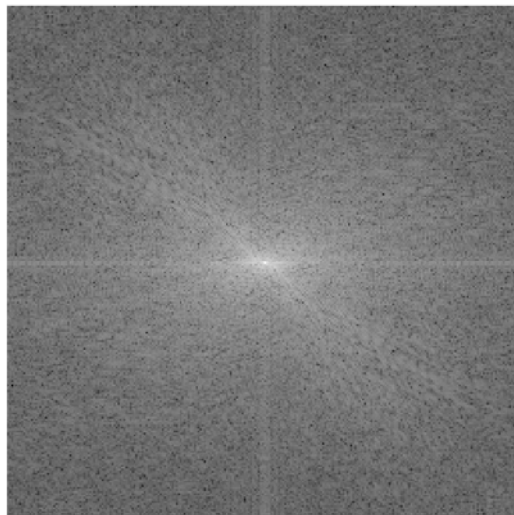


Figura 2.8 – FFT da Figura 1.1.

A FFT opera primeiramente decompondo uma função de N pontos em várias funções compostas cada uma por apenas um ponto. O segundo passo é calcular os N espectros de frequência correspondentes às N funções obtidas. Por fim, estes N espectros são sintetizados em apenas um espectro [SMITH, 2003].

A Figura 2.9 mostra um exemplo da decomposição de funções, utilizada pela FFT. Neste exemplo, uma amostra de 16 pontos é decomposta passando por quatro estágios.

O primeiro estágio divide a amostra de 16 pontos em duas amostras consistindo cada uma por oito pontos. No segundo estágio, as duas amostras obtidas são divididas em quatro amostras com quatro pontos cada. No terceiro estágio as duas amostras obtidas são divididas cada uma em outras duas amostras, obtendo-se quatro amostras.

Existe aí uma proporção de $\log_2 N$ estágios requeridos nesta decomposição. Como no exemplo, a amostra de 16 pontos, onde $16 = 2^4$, necessita de quatro estágios de divisão das amostras. Logo, uma amostra com 512 pontos, onde $512 = 2^9$, necessita de sete estágios de divisão. Uma amostra de 4096 pontos (2^{12}) necessita de 12 estágios e assim por diante [SMITH, 2003].

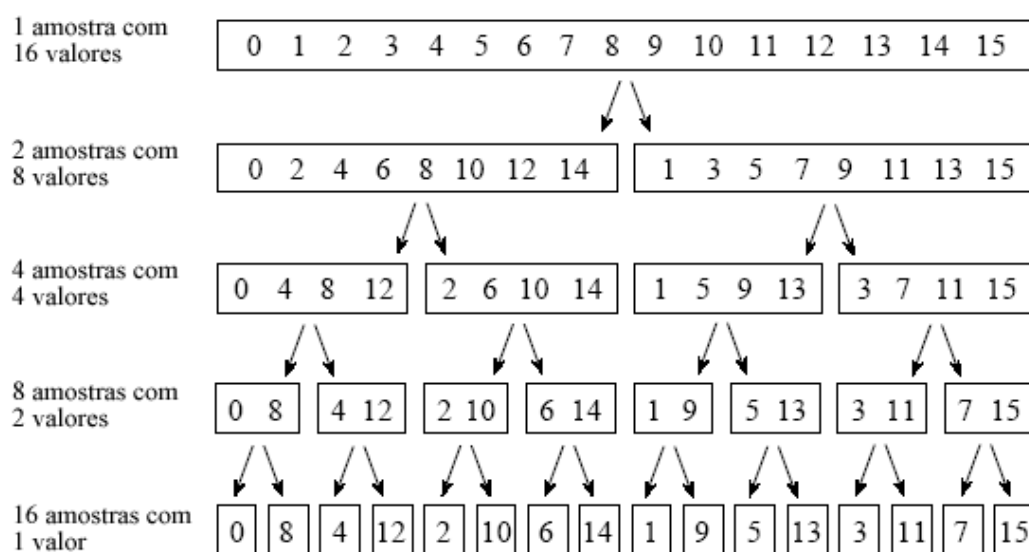


Figura 2.9 – Decomposição da FFT.

Fonte: [SMITH, 2003], com adaptações.

Ainda citando como exemplo a Figura 2.9, é necessário se obter o espectro de frequência da amostra. No último estágio, existem 16 amostras onde cada uma contém apenas um ponto.

O espectro de frequência de um sinal de um ponto é igual a ele mesmo. Fonte: [SMITH, 2003].

Portanto, cada amostra do último estágio, contendo um ponto apenas, pode ser entendida como sendo o espectro de frequência de cada ponto e não mais como um ponto da amostra no domínio espacial.

O próximo passo da FFT é combinar os N espectros de frequência calculados na ordem reversa exata à original. Ao final, têm-se o espectro de frequência da amostra inicial, que, ao sofrer a aplicação da FFT inversa apresenta-se em sua forma espacial, permitindo a visualização da imagem trabalhada.

2.4.3 – Resposta à Frequência

Utilizando-se a DFT em um sistema de tratamento de imagens, cada imagem de entrada pode ser então expressa por um conjunto de ondas senos ou cossenos, cada uma com

sua frequência e amplitude. Logo, a imagem de saída pode ser representada obedecendo esta mesma forma, após o processamento que lhe dará origem [SMITH, 2003].

Isto significa que um sistema linear de imagem pode ser descrito pelas mudanças ocorridas na fase e amplitude de suas ondas seno ou cosseno, que passaram através de tal sistema [SMITH, 2003]. Essa informação pela qual o sistema pode ser descrito é chamada no domínio espacial de resposta ao impulso, dada por $h[x,y]$. Em processamento de imagens, a resposta ao impulso é denominada PSF (“*Point Spread Function*”). Sua FT (no domínio da frequência) é a resposta à frequência e é dada por $H[u,v]$.

A resposta à frequência de um sistema é a Transformada de Fourier de sua resposta ao impulso. Fonte: [SMITH, 2003].

A Figura 2.10 ilustra a relação existente entre a resposta ao impulso e a resposta à frequência. Em uma comparação de um sistema no domínio espacial e no domínio da frequência, visualiza-se como ocorre a aplicação da afirmação citada no parágrafo anterior.

Neste trabalho será utilizada a operação em domínio da frequência, pela facilidade de cálculos e pelas vantagens já citadas de se trabalhar com a FT.

Os sistemas no domínio espacial são descritos pela convolução – que não será abordada neste trabalho, por não ser o objetivo aplicado – que é representada pelo símbolo asterisco (*). Da Figura 2.10, tem-se para o domínio espacial que a imagem degradada é o resultado da convolução entre a imagem original e a resposta ao impulso do sistema:

$$g[x,y] = f[x,y] * h[x,y]. \quad (2.5)$$

No domínio da frequência, a operação semelhante à realizada na Equação (2.5) é a multiplicação da FT da imagem original com a resposta à frequência do sistema:

$$G[u,v] = F[u,v] \cdot H[u,v]. \quad (2.6)$$

Na Figura 2.10 verifica-se ainda que a DFT atua convertendo as funções do domínio espacial para o domínio da frequência. Já a inversa da DFT (IDFT), como o próprio nome diz, faz a operação inversa.

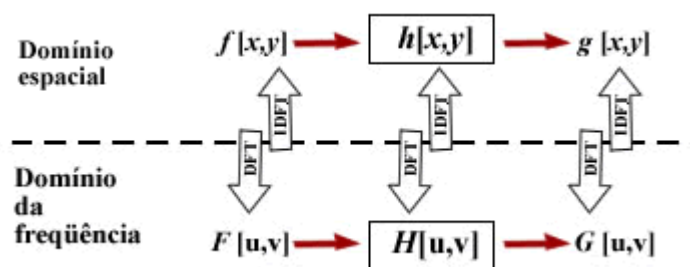


Figura 2.10 – Comparação entre sistemas no domínio espacial e da frequência.

Fonte: [SMITH, 2003], com adaptações.

2.5 –Imagem utilizada no projeto

Decidiu-se pelo uso de uma imagem monocromática em tons de cinza, já que possui como principal atributo a intensidade de cada *pixel*. A intensidade de cada *pixel* é um fator importante, como já foi citado, no momento da amostragem e quantização da imagem.

A imagem monocromática em tons de cinza é bastante utilizada em PDI, por possibilitar uma maior facilidade no processamento dos cálculos, já que é representada por uma função bidimensional $f[x,y]$. Além disso, o valor de $f[x,y]$ em cada ponto é proporcional ao tom de cinza desse ponto.

O termo “tom de cinza” geralmente é utilizado para descrever intensidades monocromáticas de luz, já que sua abrangência vai do preto, passa por diversos níveis de cinza e finalmente atinge a cor branca [GONZALEZ, 2002].

Em uma imagem monocromática em 256 tons de cinza, cada *pixel* é armazenado em um *byte*⁸, tendo seu valor definido entre o intervalo de 0 a 255, como já citado. Logo, a imagem em 256 tons de cinza requer 8 bits⁹ por *pixel* para definir as diferentes tonalidades de cinza. Então, com 8 bits por *pixel*, tem-se 256 tons de cinza.

Assim sendo, a utilização de uma imagem monocromática em 256 tons de cinza, têm como facilidades:

- Maior velocidade de processamento, já que cada *pixel* corresponde a apenas 1 *byte*;
- Menor utilização de memória para armazenamento, já que ocupa menor espaço que uma imagem colorida;

⁸ Byte: conjunto de 8 bits.

⁹ Bit (“*Binary Digit*”): menor unidade de informação de um computador. Assume valores 0 e 1.

- Facilidade de computação dos cálculos, já que é representada como uma função bidimensional $f[x,y]$.

2.5.1 – Características da imagem

A imagem utilizada na simulação deste trabalho possui formato JPEG (“*Joint Photographic Experts Group*”).

O JPEG é um padrão de mecanismo de compressão de imagens. É utilizado para comprimir imagens de cenas reais, ou seja, principalmente fotografias. Não é um bom padrão de arquivo para imagens vetoriais [JPEG FAQ, 2004].

O JPEG foi o formato escolhido por possuir, especialmente, as seguintes vantagens:

- Trabalha com imagens com tamanho reduzido;
- É possível aumentar ou diminuir a resolução da imagem, de acordo com o desejado;
- Facilidade na transmissão dos arquivos de imagens por redes de computadores, *e-mails*¹⁰ e demais meios eletrônicos;
- Possibilidade de maior armazenamento de imagens devido à menor alocação de memória por seu tamanho reduzido obtido pela taxa de compressão que pode ser obtida para a imagem;
- Formato de imagem bastante difundido e utilizado na *Internet*¹¹ (*sites*¹², *downloads*¹³).

A imagem utilizada possui as características apresentadas na Tabela 2.1.

¹⁰ E-mail: Abreviatura de *electronic-mail* (correio eletrônico). Em princípio, qualquer correio feito por meio de computadores é *e-mail*, mas o nome é utilizado principalmente para a comunicação via *Internet*. Fonte: [49].

¹¹ Internet: Rede mundial de computadores, que conecta simultaneamente computadores de todas as partes do mundo.

¹² Site: Palavra em inglês que significa local, lugar. Na *Internet*, designa um conjunto de páginas que representa uma pessoa, uma instituição ou uma empresa na rede. O termo *home-page* é usado para a página principal de um *site*. Fonte: [49].

¹³ Download: Transferência de um arquivo de um servidor para um cliente. Por exemplo, de um *site* da *web* para o computador de um usuário utilizando a *Internet*.

Tabela 2.1 – Características da imagem utilizada no trabalho.

Nome do arquivo: Lenna	Largura: 256 <i>pixels</i>
Extensão do arquivo: JPEG	Altura: 256 <i>pixels</i>
Escala: 256 tons de cinza	Classe: 8 bits/ <i>pixel</i>

Capítulo 3 – Restauração de imagens

3.1 –A Restauração

Existem processos que tratam a melhoria de uma imagem como uma técnica subjetiva. Nesses casos, onde a imagem sofre um realce em sua aparência, o principal fator de validação de sua qualidade final é a visualização humana. Esta técnica, denominada realce de imagens parte de procedimentos com base em aproximações progressivas dos dados obtidos com objetivo de manipular a imagem, de modo que se possa associá-la com os aspectos psicofísicos do sistema visual humano.

Já a restauração de imagens é uma técnica diferente da utilizada para um simples realce de aparência. Nela se encontra um processo objetivo, onde a imagem também tem sua aparência visualmente melhorada, mas que leva em conta os resultados quantitativos e qualitativos e numéricos obtidos.

Esta é uma técnica de reconstrução ou recuperação de imagens degradadas com um conhecimento prévio do fator degradante e que atua em cada ponto da imagem [GONZALEZ, 2002].

O principal fator de análise deste trabalho é o grau de reversão dos resultados obtidos com a aplicação de degradação em uma imagem.

Portanto, a informação que não se encontre mais presente na imagem degradada, não poderá ser restaurada. Esta afirmação é explicada com base no fato de que a restauração de imagens tem como objetivo reconstruir ou recuperar uma imagem que sofreu uma degradação, utilizando-se o conhecimento do fator degradante para aplicar o processo inverso em prol da reconstrução de tal imagem [GONZALEZ, 2002].

3.1.1 – A técnica da Restauração

Os principais passos na técnica da restauração são:

- Modelar os processos de degradação e de restauração;
- Aplicação de filtros para obtenção de uma imagem aproximada à original.

A Figura 3.1 ilustra os processos de degradação e restauração.

O processo de degradação é modelado por uma função de degradação $h[x,y]$ que juntamente com um ruído aditivo $\eta[x,y]$, opera sob a imagem original $f[x,y]$, produzindo uma imagem degradada $g[x,y]$.

Como o próprio nome diz, o ruído aditivo é um sinal adicionado ao processo de degradação. Já a função de degradação $h[x,y]$ atua ocasionando o embaçamento na imagem. Logo, a degradação aplicada na simulação deste trabalho utiliza uma função de degradação que provoca um embaçamento na imagem e em seguida, aplica-se sobre esta imagem um ruído aditivo.

Tendo-se o conhecimento da imagem degradada, do fator degradante e do ruído aditivo, o objetivo da restauração é então obter $\hat{f}[x,y]$ que é a imagem aproximada à original.

Essa estimativa deve ser a mais próxima possível da imagem original. Logo, quanto mais exato for o conhecimento de $h[x,y]$ e $\eta[x,y]$, mais próxima a imagem $\hat{f}[x,y]$ será de $f[x,y]$ [GONZALEZ, 2002].

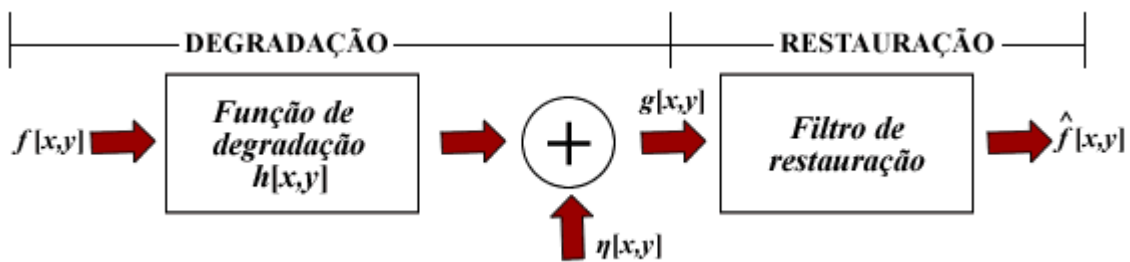


Figura 3.1 – Modelo do processo de degradação e restauração de uma imagem.

Fonte: [GONZALEZ, 2002], com adaptações.

3.2 – Modelo de Degradação

O modelo de degradação oferece uma visão geral de como ocorre a inserção de um fator degradante em uma imagem.

A Figura 3.2 demonstra o modelo de degradação de um sistema de restauração de imagens.

Representando-se matematicamente a ilustração dada na Figura 3.2, no domínio da frequência, tem-se que [GONZALEZ, 2002]:

$$G[u, v] = H[u, v]F[u, v] + N[u, v],$$

onde os termos $G[u, v]$, $H[u, v]$, $F[u, v]$ e $N[u, v]$ são as representações das FT's (espectros de frequência) de $g[x, y]$, $h[x, y]$, $f[x, y]$ e $\eta[x, y]$, respectivamente.



Figura 3.2 – Modelo de degradação.

Neste trabalho, primeiramente, foi adicionado o embaçamento à imagem e, em seguida, adicionou-se o ruído. Para isso, a equação utilizada a princípio não leva em consideração o ruído, sendo representada por:

$$G[u, v] = H[u, v]F[u, v]. \quad (3.2)$$

Sabe-se de registros científicos, que o processo inverso da degradação pode ser derivado do próprio modelo de degradação. Depende-se da sua profundidade de detalhamento.

3.2.1 – Embaçamento (“*Blurring*”)

Primeiramente tratando sobre o embaçamento, pode-se entendê-lo como sendo um desajuste comum que limita a acurácia das imagens. Uma das razões para o embaçamento em imagens são os movimentos, vibrações da câmera durante a aquisição da imagem, falta de foco. Além disso, o embaçamento pode ocorrer quando objetos se movem mais que um *pixel* no plano da imagem durante a aquisição [JÄHNE, 2002].

O embaçamento, é definido pela função denominada PSF (“*Point Spread Function*”), representada por $h[x, y]$.

Na prática, a obtenção dos parâmetros exatos da função de degradação ou PSF, $h[x,y]$, não é tarefa das mais fáceis. Este fato decorre devido à degradação real poder ser resultado de várias PSF's simultaneamente [WEN, 2002].

Em grandes empresas que estudam os processos de restauração de imagens, existem catalogados diversos tipos de PSF's (embaçamento por movimento unidimensional, embaçamento por movimento bidimensional, etc.). Logo, a prática mais comum é, portanto, fazer a escolha do modelo apropriado de PSF para a imagem a ser restaurada.

Neste trabalho, será aplicada uma degradação por embaçamento juntamente com ruído aditivo. Optou-se por não se fazer tratamentos separados para restauração de embaçamento e de ruído, pois:

- Em caso de embaçamento, a equação que descreve o filtro Wiener acaba por igualar-se à do filtro inverso, apresentando os mesmos resultados;
- Observa-se na prática que o padrão encontrado de imagens degradadas ocorre por degradações simultâneas, ou seja, mais de um tipo de degradação por imagem.

Dentre os modelos de PSF's existentes, no trabalho será utilizado o embaçamento bidimensional por movimento. Como citado mais acima, a função PSF para embaçamento bidimensional já existe catalogada para os tipos de PSF's existentes. Sua equação é dada por:

$$h[m,n] = \frac{1}{L^2} \quad (3.3)$$

onde L é a distância percorrida pelo movimento [WEN, 2002], em *pixels*. Neste trabalho, a simulação foi realizada com um valor aleatório igual a 11 (onze) para a variável L. Quanto maior for o valor definido para L, maior será o efeito do embaçamento na imagem.

O resultado obtido com a inserção da degradação por embaçamento pode ser observado na Figura 3.3.



Figura 3.3 – Imagem original embaçada.

3.2.2 – Ruído aditivo

Juntamente com o embaçamento, foi utilizado o ruído aditivo do tipo gaussiano.

Os ruídos em imagens são adicionados principalmente durante a aquisição (digitalização) e/ou transmissão de tais imagens e a performance dos sensores dos sistemas de imagens pode ser afetada por uma variedade de fatores como condições climáticas, vida-útil dos sensores, qualidade do equipamento, interferências nos canais de transmissão, etc. [GONZALEZ, 2002].

Ruídos em sistemas de imagens podem ser aditivos ou multiplicativos. No caso aditivo, matematicamente descreve-se por uma adição do mesmo ao sistema, como pôde ser observado na Equação (3.1).

Ruídos gaussianos são freqüentemente utilizados na prática, devido à sua facilidade de tratamento matemático tanto no domínio espacial quanto no domínio da freqüência. O ruído gaussiano também é chamado de ruído normal [GONZALEZ, 2002].

A função que descreve a função de densidade de probabilidade gaussiana de uma variável aleatória z é dada pela Equação (3.4).

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}}, \quad (3.4)$$

onde z representa o nível de cinza, μ é a média dos valores de cinza e σ é seu desvio padrão [GONZALEZ, 2002]. O símbolo σ^2 é chamado variância, no caso, de z .

Os resultados obtidos com a inserção das degradações com diferentes valores de SNR, variando de 5dB a 100dB, podem ser observados no ANEXO B deste trabalho.

3.2.3 – Definição de $h[x,y]$

A matriz de $h[x,y]$ é dada na forma da Equação (3.6). A Equação descreve o formato da matriz quadrada de dimensão $M \times N$, onde $M = N$.

Essa matriz deve possuir a mesma dimensão da imagem a ser restaurada.

$$h = \begin{bmatrix} h_0 & h_{M-1} & h_{M-2} & \dots & h_1 \\ h_1 & h_0 & h_{M-1} & \dots & h_2 \\ h_2 & h_1 & h_0 & \dots & h_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{M-1} & h_{M-2} & h_{M-3} & \dots & h_0 \end{bmatrix} \quad (3.6)$$

Neste trabalho, foi utilizada uma imagem de dimensão 256x256 pixels. Sendo assim, $h[x,y]$ foi definida possuindo a mesma dimensão. A Figura 3.4 apresenta como é definida a matriz que define a função PSF no código fonte da simulação deste trabalho.

No caso, a função PSF tem como objetivo causar o embaçamento, e é definida pelo nome “hc” que armazena uma matriz quadrada de dimensão 256x256 *pixels*.

Os passos são descritos a seguir:

- Define-se uma matriz quadrada de dimensão 256x256 *pixels*, preenchida com zeros e armazenada em “hc”;
- De acordo com a Equação (3.3), apresentada no tópico 3.3 deste trabalho, define-se o fator degradante da matriz da PSF de embaçamento bidimensional por movimento. O fator definido foi $h[m,n] = \frac{1}{11^2}$.
- Aplica-se, então, o fator definido para a PSF nos elementos da matriz, para que possam interagir sobre imagem provocando o embaçamento.

3.3 – Modelo de Restauração

A restauração de imagens parte do princípio de que a degradação é conhecida.

A Figura 3.4 apresenta o modelo de restauração de uma imagem digital.

Verifica-se que a partir de uma imagem degradada $g[x,y]$ e com a utilização do filtro de restauração desejado, é possível se obter uma imagem $\hat{f}[x,y]$ aproximada à original $f[x,y]$.



Figura 3.4 – Modelo de restauração.

Esse processo de restauração pode ser repetido até que se encontre um resultado satisfatório [WEN, 2002].

Capítulo 4 – Filtragem no domínio da frequência

4.1 – Conceito

A filtragem no domínio da frequência é dita direta. Segue os seguintes passos, de acordo com a Figura 4.1:

- Tem-se a imagem degradada $g[x,y]$;
- É computada a DFT da imagem degradada $g[x,y]$, obtendo-se $G[u,v]$;
- Aplica-se a função $H[u,v]$ do filtro desejado sobre $G[u,v]$, obtendo-se $\hat{F}[u,v]$;
- Computa-se a DFT inversa de $\hat{F}[u,v]$ para se obter a imagem aproximada à original, $\hat{f}[x,y]$.

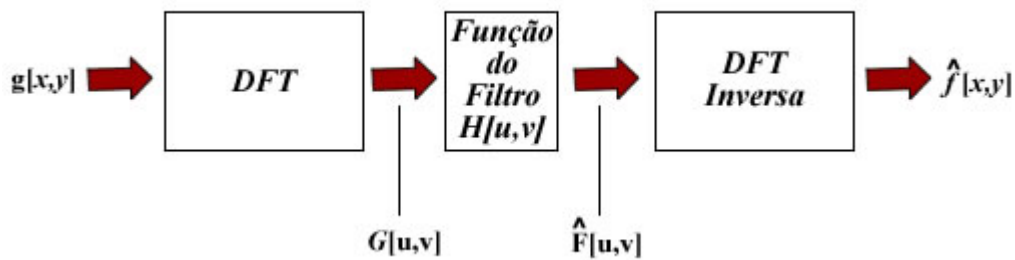


Figura 4.1 – Passos básicos da filtragem no domínio da frequência.

Fonte: [GONZALEZ, 2002], com adaptações.

A imagem filtrada é, portanto, obtida fazendo-se a DFT inversa de $\hat{F}[u,v]$:

$$\hat{f}[x,y] = \mathfrak{T}^{-1}[\hat{F}[u,v]]. \quad (4.1)$$

Resumindo o processo, pode-se dizer que a filtragem é baseada na modificação da imagem degradada a partir de sua DFT e da função do filtro utilizado. Fazendo-se a inversa do resultado obtido tem-se, por fim, a imagem de saída, filtrada e aproximada à original [GONZALEZ, 2002].

4.2 - Filtragem inversa

O filtro inverso é uma aplicação simples para se obter $\hat{f}[x,y]$, que é a imagem aproximada à original. Tem como objetivo aplicar a função inversa (ou mais aproximada possível) à função de degradação que foi aplicada à imagem que se encontra degradada.

A Figura 4.2 apresenta uma visão geral da filtragem inversa. Considerando-se, no domínio da frequência, $H^I[u,v]$ como sendo a função do filtro de restauração, obtém-se da Figura 4.2, a seguinte relação:

$$G[u,v] = \hat{F}[u,v]H[u,v] \quad (4.2)$$

\Downarrow

$$\hat{F}[u,v] = \frac{G[u,v]}{H[u,v]} \quad (4.3)$$



Figura 4.2 – Filtro inverso.

E na presença de ruído aditivo, tem-se [GONZALEZ, 2002]:

$$\hat{F}[u,v] = F[u,v] + \frac{N[u,v]}{H[u,v]}, \quad (4.4)$$

lembrando-se que $F[u,v] = \frac{G[u,v]}{H[u,v]}$.

4.2.1 – Filtragem inversa – procedimento prático.

A função $H[u,v]$ pode apresentar alguns pontos com valores iguais a zero. Para esses pontos de $H[u,v]$, existe o fato de que um problema computacional pode ocorrer, devido à divisão por zero.

Para isso, a solução utilizada na prática é limitar as frequências do filtro a valores próximos à origem de seu espectro [GONZALEZ, 2002], onde o valor de $H[u,v]$ é suficientemente grande [WEN, 2002].

Por exemplo, observando-se o espectro de frequências apresentado na Figura 2.8, verifica-se a concentração de frequências mais altas próximas à origem do plano, se estendendo pelas divisões dos quadrantes.

Portanto, é necessário que seja fixado um parâmetro para limitar as frequências do filtro. No caso, foi definido o parâmetro “*thresh*”. Tal parâmetro foi assim definido, pois possui comportamento semelhante ao do *thresholding*.

O fundamento básico do “*thresholding*” é de que uma imagem seja composta de elementos de cor claros e de um fundo (“*background*”) escuro, define-se um parâmetro denominado “*thresholding*” de modo que os *pixels* da figura principal e do background possam ser manipulados com maior e/ou menor brilho. A partir de certo ponto definido pelo “*thresholding*”, manipulam-se os *pixels* de forma a se obter maiores ou menores frequências de cor para o fundo ou para a imagem.

No caso, o parâmetro “*thresh*” servirá como fator limitante do filtro, fixando um valor de $H[u,v]$ próximo à origem de modo que não ocorra a complicação citada para valores de H muito pequenos e/ou iguais a zero na equação que define a filtragem inversa. Com isso, limita-se um $H[u,v]$ que trabalhe com frequências mais altas, evitando-se o problema computacional da divisão por zeros.

As simulações foram feitas, portanto, com três valores fixados para o parâmetro “*thresh*”: 0.000001, 0.05 e 0.1.

4.2.2 – Vantagens e desvantagens

A filtragem inversa é ótima para restaurar imagens que sofreram degradação por embaçamento. Mas no caso de imagens com ruído, verifica-se pela Equação (4.4) que, para valores de $H[u,v]$ muito pequenos ou próximos de zero, o ruído poderá dominar a imagem ocasionando falha de restauração e/ou um problema computacional de cálculo com divisão por zero.

Com isso, para a implementação do filtro inverso é necessário dedicar atenção especial a um fator limitante de $H[u,v]$, no caso foi definido o parâmetro denominado “*thresh*”, para limitar as frequências de $H[u,v]$ a pontos próximos à origem de seu espectro de frequências, onde há concentração de altas frequências, como já visto anteriormente. Este é um fator que

provoca uma desvantagem para a aplicação deste tipo de filtragem, já que caso este fato não seja abordado no desenvolvimento da aplicação, o filtro poderá não ser satisfatório.

Apesar disto, o filtro inverso é uma maneira rápida e de fácil implementação na restauração de imagens. Tal fato explica sua ampla utilização no campo de PDI.

Por ser simples e de fácil implementação, o código fonte pôde ser feito de forma clara, enxuta, simples e que não necessita de exagerado esforço computacional.

4.3 - Filtragem Wiener

A filtragem inversa é bastante sensível a ruídos. Com isso, a aproximação para a redução da degradação envolve a elaboração de algoritmos para cada tipo de degradação e, se for o caso, a combinação entre eles. A filtragem Wiener encontra um ótimo equilíbrio entre a filtragem inversa e redução de ruído [WANG, 1999].

O filtro Wiener é associado a Norbert Wiener e Andrei N. Kolmogorov, cujos relatos datam de 1940 [THESSALONIKI, 2003].

O filtro Wiener permite a restauração de uma imagem baseada nas frequências espectrais. É considerado um método otimizado já que minimiza o erro médio quadrático no processo de filtragem e suavização do ruído.

Com essa técnica, assume-se que o espectro de potência da imagem original e o espectro de potência da imagem com ruído são conhecidos e busca-se minimizar a diferença entre a imagem restaurada e a original, considerando-se, portanto os efeitos do ruído.

Então, seu objetivo é minimizar o erro médio quadrático, dado por [JIANG, 2003]:

$$|e| = |f[x, y] - \hat{f}[x, y]| \quad (4.5)$$

onde e é o erro médio quadrático, “E” é o valor esperado do argumento, $f[x, y]$ é a imagem original e $\hat{f}[x, y]$ é a imagem restaurada aproximada à original.

Considerando-se que $\hat{f}[x, y]$ é derivada de uma operação de filtragem linear “w” sobre a imagem degradada $g[x, y]$, tem-se [JIANG, 2003]:

$$\hat{f}[x, y] = w[x, y]g[x, y]. \quad (4.6)$$

Aplicando-se a DFT na Equação (4.6), temos os seguintes termos: $\hat{F}[u, v]$, $W[u, v]$ e $G[u, v]$.

Substituindo-se a Equação (4.6) na Equação (4.5), e em seguida aplicando-se a DFT, obtém-se [JIANG, 2003]:

$$|e|^2 = |F[u, v] - \hat{F}[u, v]|^2 \quad (4.7)$$

$$= |F[u, v] - W[u, v]G[u, v]|^2 \quad (4.8)$$

$$= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (W[u, v]G[u, v] - F[u, v])(W[u, v]^* G[u, v]^* - F[u, v]^*) \quad (4.9)$$

Fazendo-se a diferenciação da Equação (4.9), em relação ao termo $W[u, v]^*$, tem-se [JIANG, 2003]:

$$G[u, v]^* (W[u, v]G[u, v] - F[u, v]) = 0, \quad (4.10)$$

para $u = 0, \dots, M-1$ e $v = 0, \dots, N-1$.

Isolando-se o termo $W[u, v]$, para dar início à definição da fórmula que descreve o filtro Wiener na presença de ruído $N[u, v]$:

$$W[u, v] = \frac{G[u, v]^* F[u, v]}{|G[u, v]|^2} \quad (4.11)$$

$$= \frac{(H[u, v]^* F[u, v]^* + N[u, v]^*) F[u, v]}{|H[u, v]|^2 |F[u, v]|^2 + 2 \operatorname{Re}[H[u, v] F[u, v] N[u, v]^*] + |N[u, v]|^2} \quad (4.12)$$

Da propriedade de correlação discreta, tem-se [JIANG, 2003]:

$$|F[u, v]|^2 = S_f \quad \text{e} \quad |N[u, v]|^2 = S_n, \quad (4.13)$$

em que S_f e S_n são os espectros de potência de $F[u, v]$ e $N[u, v]$, respectivamente.

Como a restauração de imagens assume que o ruído $N[u, v]$ e a imagem $F[u, v]$ não possuem correlação, pode-se dizer que uma relação R entre as duas funções é dada por:

$$R_{fn} = 0. \quad (4.14)$$

Utilizando-se ainda a propriedade de correlação discreta (similar á convolução circular em domínio do espaço), que diz que:

$$F[R_{fg}] = F[f[x,y]]F[g[x,y]^*], \quad (4.15)$$

onde R_{fg} significa a auto-correlação das funções $f[x,y]$ e $g[x,y]$ com suas DFT's. Tal auto-correlação é chamada de espectro de potência de $f[x,y]$ e $g[x,y]$ denotado por S_f e S_g , respectivamente [JIANG, 2003].

Portanto, tem-se:

$$F[R_{fn}] = F[u,v]N[u,v]^* = 0. \quad (4.16)$$

Então, simplificando-se a Equação (4.12), tem-se:

$$W[u,v] = \left[\frac{H[u,v]^* S_f[u,v]}{|H[u,v]|^2 S_f[u,v] + S_n[u,v]} \right] G[u,v] \quad (4.17)$$

$$= \left[\frac{H[u,v]^*}{|H[u,v]|^2 + \frac{S_n[u,v]}{S_f[u,v]}} \right] G[u,v] \quad (4.18)$$

Substituindo-se $|H[u,v]|^2 = H^*[u,v]H[u,v]$, na Equação (4.18), tem-se:

$$= \left[\frac{1}{H[u,v]} \frac{|H[u,v]|^2}{|H[u,v]|^2 + \frac{S_n[u,v]}{S_f[u,v]}} \right] G[u,v]. \quad (4.19)$$

É interessante ressaltar o fato de que na ausência de ruído, $S_\eta[u,v]=0$, o filtro Wiener torna-se um filtro inverso perfeito. Para demonstrar este fato, verifica-se da Equação (4.19), que na ausência de ruído, não há espectro de potência do ruído, portanto o termo $S_\eta[u,v]/S_f[u,v]$ iguala-se a zero:

$$W[u,v] = \left[\frac{1}{H[u,v]} \frac{|H[u,v]|^2}{|H[u,v]|^2 + 0} \right] G[u,v] \quad (4.20)$$

Simplificando a Equação (4.20), tem-se:

$$W[u,v] = \left[\frac{1}{H[u,v]} \right] G[u,v] \quad (4.21)$$

$$= \left[\frac{G[u,v]}{H[u,v]} \right] \quad (4.22)$$

Comparando-se as Equações (4.3) e (4.22), verifica-se que na ausência de ruído os dois filtros se igualam, obtendo-se, portanto, os mesmo resultados.

4.3.1 – Filtragem Wiener – procedimento prático.

Para implementar o filtro Wiener na prática, é preciso estimar o espectro de potência da imagem original e do ruído aditivo.

Na simulação foi definido um parâmetro “sigmaN” que define a variância do ruído e é dado por:

$$sigmaN = \sqrt{\frac{(sigmaG)^2}{10 \left(\frac{N}{10} \right)}} \quad (4.23)$$

onde:

N = valor do desvio padrão aplicado à imagem;

sigmaG = desvio padrão de $G[u,v]$;

Por exemplo, para uma imagem degradada com SNR = 5dB, tem-se:

$$\sigma_5 = \sqrt{\frac{(\sigma_G)^2}{10 \left(\frac{5}{10}\right)}} \quad (4.24)$$

O parâmetro “sigmaN” é necessário para se definir o espectro de potência do ruído, dado por:

$$S_n = (\sigma_N)^2 \quad (4.25)$$

E o espectro de potência da imagem original é dado por:

$$S_f = \frac{\text{abs}(F(u,v))^2}{N^2} \quad (4.26)$$

onde:

abs = valor absoluto;

$F(u,v)$ = DFT da imagem original $f(x,y)$;

N = dimensão da imagem quadrada, exemplo: $N=256$ para imagem de dimensão 256×256 pixels.

Em seguida aplica-se a fórmula do filtro Wiener definida na Equação (4.19).

Para que a imagem possa ser apresentada visualmente, aplica-se a DFT inversa sobre a imagem $\hat{F}[u,v]$ resultante da filtragem.

4.3.2 – Vantagens e Desvantagens.

O filtro Wiener funciona bem para uma imagem cujas propriedades são conhecidas, como a função PSF, o tipo de degradação e os espectros de potência.

Além disso, tem controle sobre o erro já que minimiza o erro médio quadrático. Na prática, possui aplicação direta sobre a imagem degradada. Um problema importante na implementação do filtro Wiener é que a aproximação da função PSF $H[u,v]$ e dos espectros de potência da imagem e do ruído, pode não ser tão precisa [THESSALONIKI, 2003], portanto, antes da restauração deve-se analisar se esses valores são conhecidos ou facilmente obtidos.

Uma das maiores qualidades do filtro Wiener, é o fato de trabalhar com o mínimo erro médio quadrático. Isso significa que os resultados obtidos com filtragem Wiener terão uma simplificação de qualquer aspecto degradante na imagem, os quais o olho humano não é capaz de perceber, mesmo com algum ligeiro ruído ainda presente [RINTOUL, 2000].

4.4 Motivos que levaram a escolha dos métodos de restauração

Os métodos de restauração escolhidos são dois tipos de filtragem bastante estudados e comuns na área de restauração de imagens.

Foram escolhidos dois métodos de filtragem (inversa e Wiener) com o objetivo de delimitar o escopo do projeto apresentado na Proposta de Projeto Final. Além disso, tal delimitação atende ao tempo disponibilizado para o cumprimento da Proposta.

As técnicas de filtragem utilizadas neste trabalho possuem vasta documentação bibliográfica.

Como citado no decorrer do trabalho, a implementação física dos códigos em Matlab é facilitada pela simplicidade da interpretação teórica, já que o Matlab possui recursos especiais para tratamento de imagens, como a “*toolbox*”¹⁴ de PDI.

¹⁴ Toolbox: biblioteca de funções do Matlab com recursos referentes a determinado tipo de aplicação.

Capítulo 5 – Simulação

5.1 – Amostra

A imagem utilizada nos capítulos seguintes, para a realização de simulações e visualização dos resultados, será monocromática conforme já citado e apresentado no Capítulo 1, Figura 1.1 e apresentada novamente a seguir na Figura 5.1.



Figura 5.1 – Imagem Original.

5.2 – Configuração do ambiente de homologação

Foi utilizado um computador pessoal, de uso doméstico, com as seguintes configurações:

- Processador AMD Athlon (TM) XP 1.34 GHz¹⁵;
- 256 MB de memória RAM¹⁶.

Softwares utilizados:

- Sistema Operacional Windows XP Professional Versão 2002;
- Macromedia Fireworks MX 2004;
- Corel Draw v. 11;
- Matlab v. 6.5;
- Microsoft Excel 2002.

¹⁵ GHz: Abreviação de GigaHertz. Corresponde a 1.000.000.000 Hz. Unidade de medida de velocidade interna do processador.

¹⁶ RAM: “Radom-Access Memory”. Memória de armazenamento temporário.

5.3 – Métodos de análise

Qualquer método de tratamento de imagens pode resultar em perdas importantes de informações ou da qualidade das imagens trabalhadas.

Para isso, existem técnicas de medidas de avaliação de desempenho das simulações feitas. Portanto, deve se possuir definidas quais serão as maneiras que servirão como avaliadoras da QoS (*“Quality of Service”*) – qualidade de serviço – obtida e desejada com o projeto proposto. Com isso, questões como: limitações do procedimento e técnicas adotados, qualidade dos resultados obtidos e viabilidade da aplicação, podem ser mais bem analisados.

Neste trabalho, foram utilizados dois métodos para análise dos resultados obtidos com as simulações práticas. Os dois métodos foram escolhidos pela sua praticidade computacional e por terem facilidade de tratamento matemático. Devido a estes fatores, é prática comum que sejam utilizados com fins de análise em PDI.

Tais métodos são: histogramas e ISNR (*“Improvement in signal to noise ratio”*).

5.3.1 – Histogramas

Para gerar os histogramas das imagens, foi utilizado o software Corel Draw versão 11. Histogramas são gráficos importantes na análise das características numéricas de imagens. Seu objetivo é permitir que se tenha uma descrição da imagem de acordo com o número de ocorrências de seus tons de cinza.

A representação de um histograma é por um gráfico de barras que aponta a quantidade de *pixels* em determinado valor de tom de cinza. Além disso, com a utilização da ferramenta Corel Draw é possível obter atributos estatísticos como a média, a mediana, o desvio padrão e o número de *pixels* da imagem, como pode ser observado na Figura 5.2.

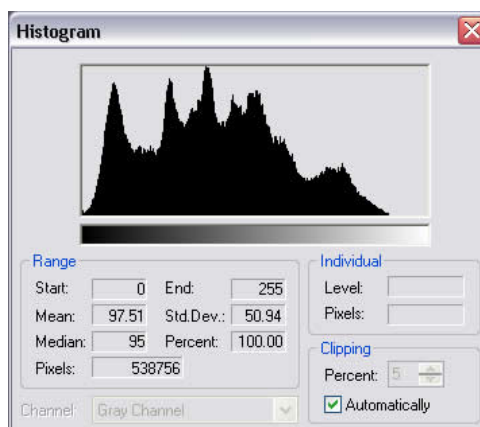


Figura 5.2 – Histograma da imagem original.

A análise de resultados pelos histogramas obtidos com a simulação, podem ser vistos no Capítulo 6 deste trabalho. Os histogramas de cada imagem utilizada na simulação podem ser vistos no ANEXO B (imagem original), ANEXO C (filtragem inversa) e ANEXO D (filtragem Wiener).

5.3.2 – SNR e ISNR

A relação sinal-ruído (SNR) representa o impacto do ruído sobre a imagem [FISHER, 2003].

A função SNR (relação sinal-ruído) é uma função de frequência, definida como:

$$SNR = 10 \log \left(\frac{\sigma_g^2}{\sigma_\eta^2} \right) \quad (5.1)$$

onde, SNR é a relação sinal-ruído em decibéis (dB), σ_g é a variância da imagem degradada a ser analisada e σ_η é a variância do ruído.

Neste trabalho, a SNR foi utilizada como parâmetro de degradação por ruído nas imagens. Por exemplo, tendo como base a imagem degradada por embaçamento, insere-se a degradação por ruído aditivo gaussiano na imagem. Coletam-se, então, os resultados para cada valor de SNR aplicado (variando entre 5dB a 100dB).

Para verificar o índice de melhoria obtido após a aplicação do filtro de restauração na imagem, é utilizado o ISNR (*“Improvement in Signal to Noise Ratio”*). Tal índice é dado por:

$$ISNR = 10 * \log_{10} \left(\frac{\sum (f(:) - g(:))^2}{\sum (f(:) - fr(:))^2} \right) \quad (4.6)$$

onde (:) representa a seleção de todos os elementos da matriz, $f(:)$ é a imagem original, $g(:)$ é a imagem degradada e $fr(:)$ é a imagem restaurada [HARE, 2004].

Os resultados obtidos com a inserção de ruído nas imagens podem ser observados no ANEXO B deste trabalho.

5.4 – Descrição do código fonte por diagramas

Esta seção tem como objetivo apresentar o código fonte utilizado na simulação do projeto por meio de diagrama, para melhor compreensão.

5.4.1 – Inserção de embaçamento na imagem

A Figura 5.3 mostra o procedimento utilizado na inserção do embaçamento na imagem original $f[x,y]$.

Descreve-se pelos seguintes passos:

- **1:** Têm-se a imagem original $f[x,y]$ e a função PSF de degradação por embaçamento $h[x,y]$;
- **2:** Aplica-se a DFT para obter a versão em domínio da frequência das duas funções;
- **3:** Utilizando-se a Equação (3.2), obtém-se a imagem degradada $G[u,v]$, a partir do produto entre $F[u,v]$ e $H[u,v]$;
- **4:** No 4º passo, aplica-se a DFT inversa para obter a imagem novamente no domínio espacial.
- **5:** Imagem obtida $g[x,y]$, degradada com embaçamento.

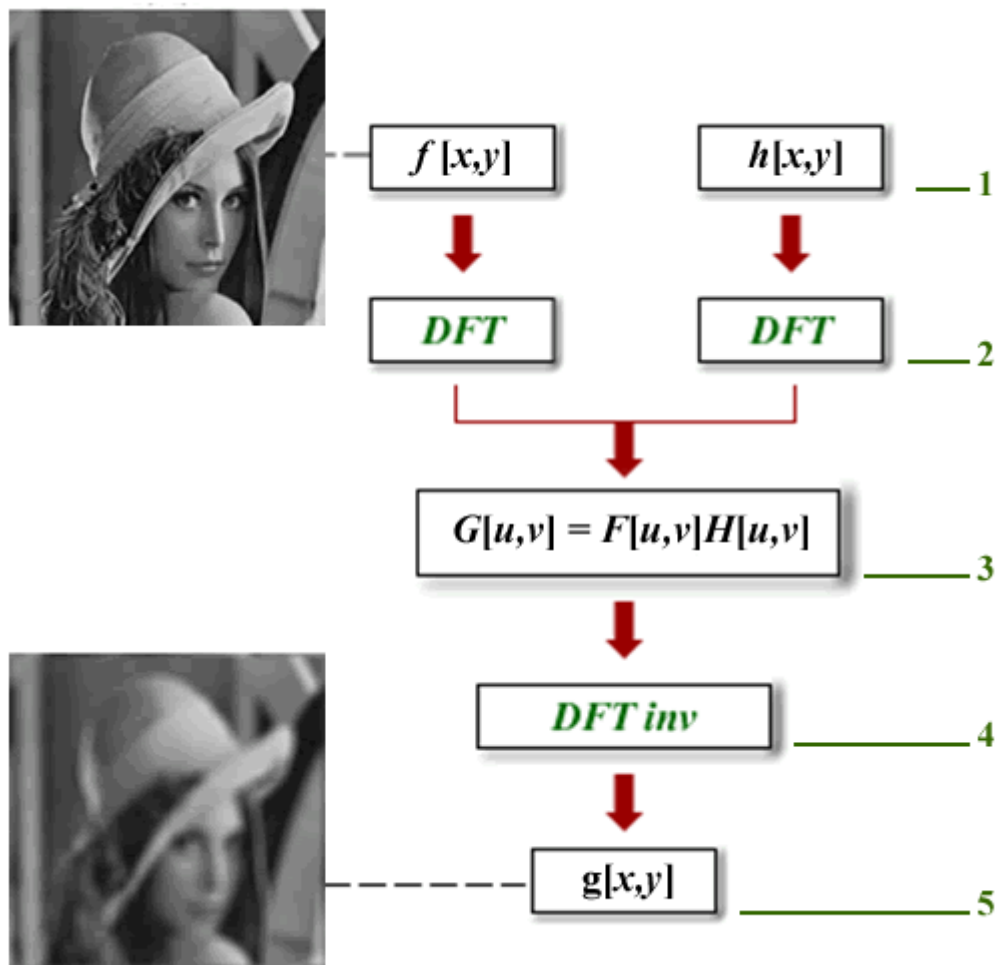


Figura 5.3 – Diagrama – Inserção de embaçamento na imagem.

O resultado da inserção de degradação por embaçamento na imagem original, pode ser visto no ANEXO B, Figura B.2.

5.4.2 – Inserção de ruído na imagem

O tipo de ruído escolhido para ser aplicado à imagem foi o tipo gaussiano.

Após a inserção do embaçamento na imagem, é adicionado o ruído. Este procedimento é mostrado na Figura 5.4 e segue os seguintes passos:

- **1:** É definido um parâmetro “sigmaN” que representa o desvio padrão do ruído. Para calcular o “sigmaN” é necessário que se tenha os parâmetros “sigmaG” e “N” já definidos. “sigmaG” representa a variância da imagem degradada e “N” representa o valor de SNR aplicado à imagem.

- 2: Após se obter o resultado de “sigmaN”, aplica-se a função “imnoise” do Matlab, que tem por finalidade adicionar um ruído à imagem “g”. Portanto, “gN” representa a imagem degradada pelo ruído de SNR = “N”. Então, o ruído é inserido na imagem “g”, a forma do ruído é “gaussian” (Gaussiana), com média “0” (zero, é padrão da função) e variância “sigmaN”.

$$\sigma N = \sqrt{\frac{(\sigma G)^2}{10 \left(\frac{N}{10}\right)}} \quad \text{— 1}$$

$$gN = \text{imnoise}(g, 'gaussian', 0, (\sigma N)^2) \quad \text{— 2}$$

Figura 5.4 –Inserção de ruído na imagem.

Os resultados obtidos com a degradação por ruído juntamente com o embaçamento, podem ser observados no ANEXO B deste trabalho.

5.4.3 – Filtragem inversa

A implementação da filtragem inversa segue os passos a seguir, apresentados na figura 5.5:

- 1: Têm-se a imagem degradada $g[x,y]$ e é conhecida a função PSF de degradação $h[x,y]$;
- 2: Aplica-se a DFT para que as funções do passo 1 possam ser trabalhadas no domínio da frequência;
- 3: Obtêm-se, então, as funções $G[u,v]$ e $H[u,v]$ no domínio Fourier (frequência). $H[u,v]$ precisa ser definido para ter um valor próximo à origem, isso é feito com o parâmetro “threshold” definido neste trabalho em três valores distintos, com finalidade de simulação comparativa: 0.000001, 0.05 e 0.1;
- 4: Aplica-se a fórmula que define a filtragem inversa, apresentada no Capítulo 4, Equação (4.3);

- 5: Sobre o resultado obtido no 4º passo, aplica-se a DFT inversa para se obter a função resultante novamente em domínio espacial, permitindo sua visualização.
- 6: Imagem resultante $\hat{f}(x, y)$;
- 7: É feita a verificação do ISNR da imagem resultante.

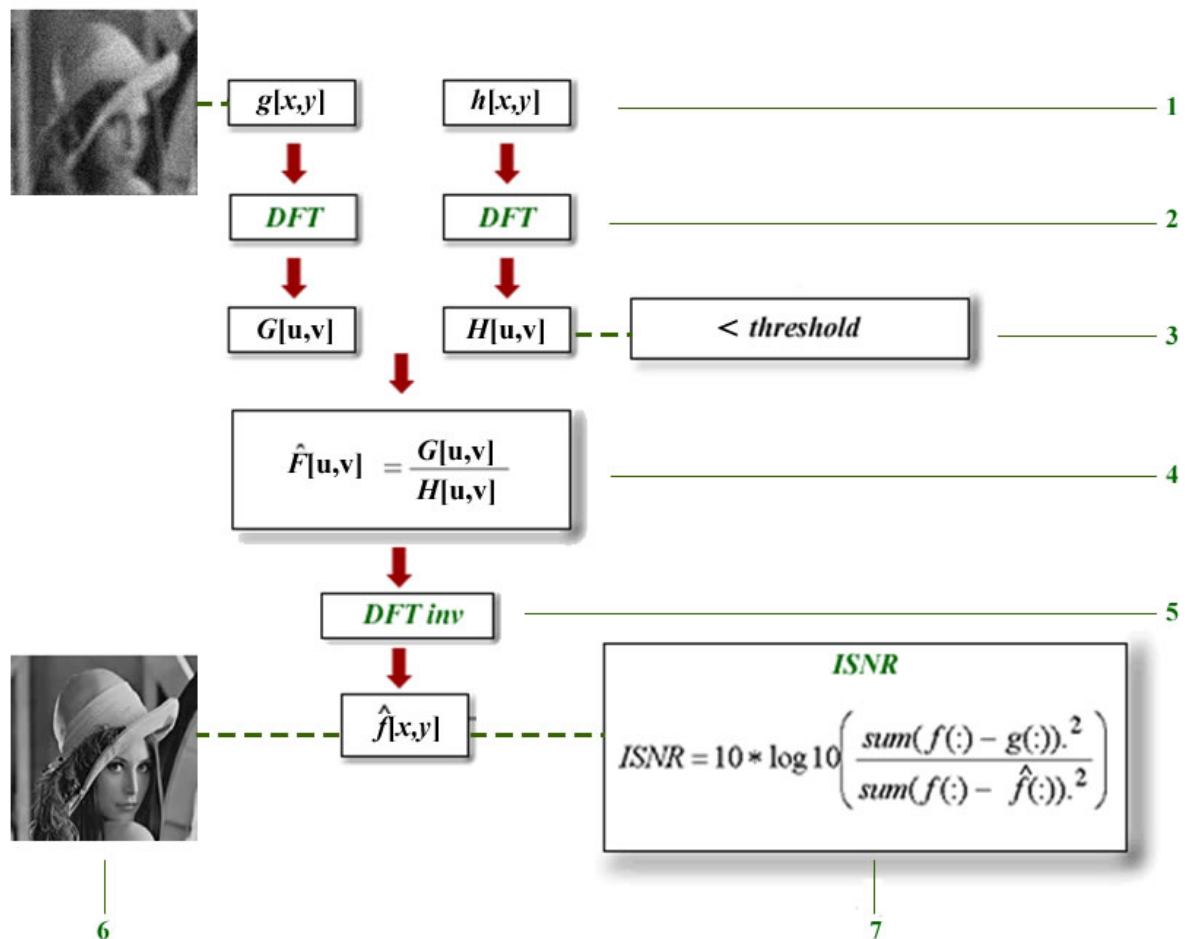


Figura 5.5 – Diagrama – Filtragem inversa.

Os resultados obtidos com a filtragem inversa podem ser observados no ANEXO C deste trabalho.

5.4.4 – Filtragem Wiener

A implementação da filtragem inversa segue os passos a seguir, apresentados na Figura 5.6:

- 1: Tem-se o conhecimento da imagem original $f[x,y]$ e que do ruído aditivo $\eta[x,y]$;

- **2:** São obtidos os espectros de potência S_f e S_n , da imagem original e do ruído respectivamente;
- **3:** Para dar início à aplicação do filtro Wiener, têm-se então os espectros de potência da imagem e do ruído, obtidos no passo 2, a função PSF de degradação $H[u,v]$ é conhecida, assim como a imagem degradada $G[u,v]$;
- **4:** Utiliza-se a fórmula que descreve o filtro Wiener, apresentada neste trabalho no Capítulo 4, Equação (4.19);
- **5:** Fazendo-se o produto entre a função já conhecida $H[u,v]$ e W – encontrada no 4º passo, tem-se a imagem resultante $\hat{F}(u,v)$ aproximada à original;
- **6:** Aplica-se a DFT inversa, sobre $\hat{F}(u,v)$ para obtê-la em domínio espacial;
- **7:** Imagem resultante $\hat{f}(x,y)$;
- **8:** É feita a verificação do ISNR da imagem resultante.

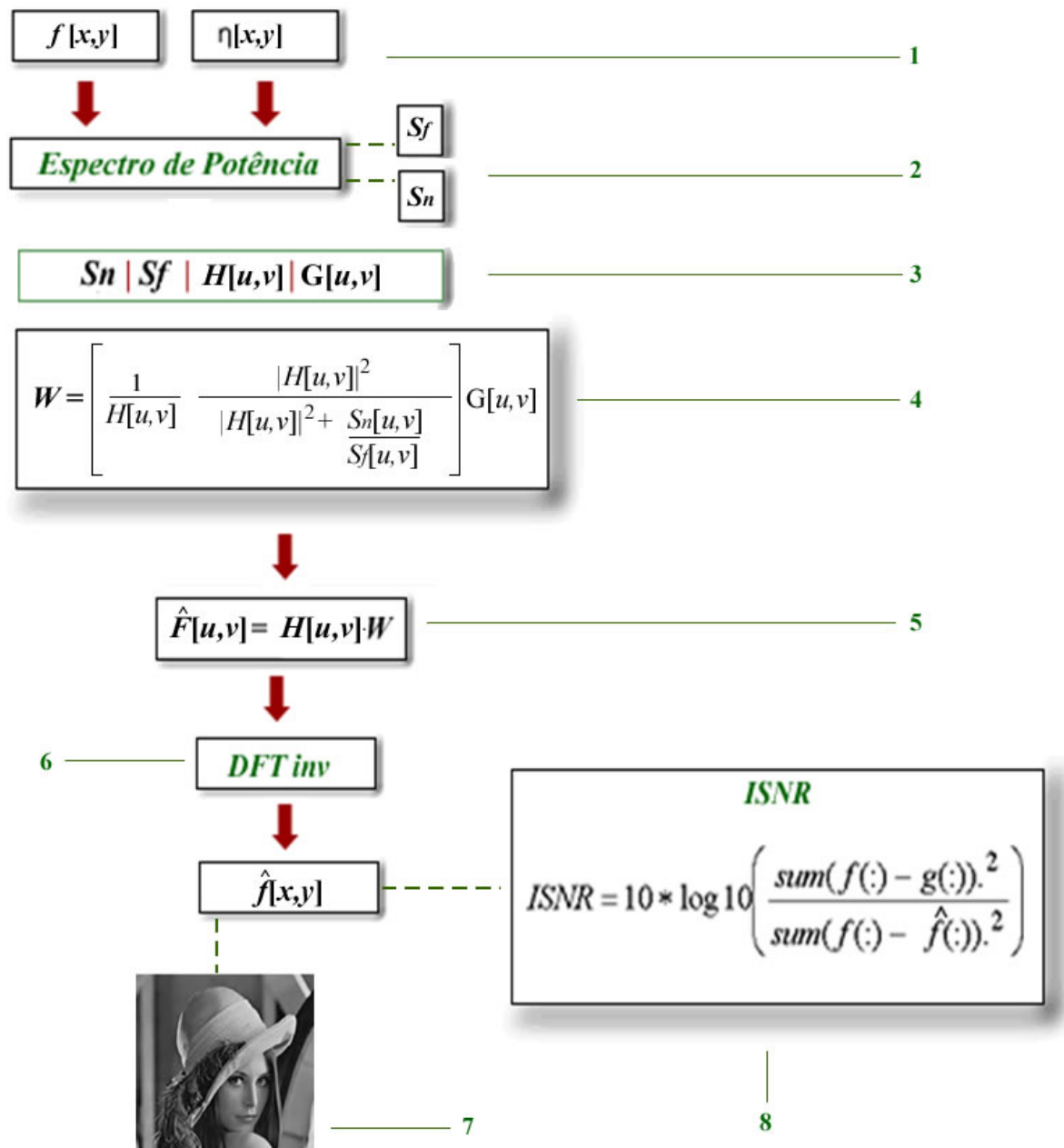


Figura 5.6 – Diagrama – Filtragem Wiener.

Capítulo 6 – Análise de Resultados

A análise dos resultados obtidos com a simulação será feita de acordo com os resultados numéricos conseguidos a partir dos métodos de análise apresentados na seção 5.3.

Os valores encontrados em cada simulação serão disponibilizados em tabelas e gráficos, para melhor visualização dos resultados.

Todos os gráficos foram feitos utilizando-se o software Microsoft Excel 2002.

6.1 - Resultados – Embaçamento e Ruído

Analisar a situação de uma imagem que sofre degradação por embaçamento e ruído, é analisar um dos casos mais comuns de degradações que ocorrem na prática. Optou-se por não apresentar o efeito do embaçamento e do ruído em tópicos separados, pois no caso de imagem sem ruído (apenas com embaçamento) o filtro Wiener iguala-se ao filtro inverso apresentado, portanto, os mesmos resultados, como foi apresentado na seção 4.3.

Nessa análise, foi utilizada a imagem de teste apresentada na Figura (1.1). Foram aplicados o embaçamento e o ruído gaussiano, com SNR variando de 5dB a 100dB (5dB, 10dB, 20dB, 30dB, 40dB, 50dB, 50dB, 60dB, 70dB, 80dB, 90dB e 100dB).

A tabela 6.1 apresenta os valores de média, desvio padrão e número de pixels para cada imagem na qual foram aplicadas as degradações, obtidas a partir de seus histogramas.

Tabela 6.1 – Resultados: imagens com degradação.

IMAGEM	SNR	MÉDIA	DESVIO PADRÃO	NÚMERO DE PIXELS
Figura B.1 (Imagem Original)	-	97,51	50,94	538756
Figura C.1	5 dB	97,81	48,87	538756
Figura C.2	10 dB	97,41	44,80	538756
Figura C.3	20 dB	97,41	43,00	538756
Figura C.4	30 dB	97,40	42,81	538756
Figura C.5	40 dB	97,40	42,79	538756
Figura C.6	50 dB	97,39	42,80	538756
Figura C.7	60 dB	97,39	42,80	538756
Figura C.8	70 dB	97,39	42,80	538756
Figura C.9	80 dB	97,39	42,80	538756
Figura C.10	90 dB	97,39	42,80	538756
Figura C.11	100 dB	97,39	42,79	538756

A partir dos dados obtidos, apresentados na Tabela 6.1, foram plotados os gráficos de média e desvio padrão, que podem ser visualizados nas Figuras 6.1 e 6.2, respectivamente.

A Figura 6.1 apresenta os valores obtidos para a média dos pixels das imagens utilizadas na simulação.

De acordo com o gráfico, nota-se que para o valor de SNR=5dB, obteve-se o maior valor de média. Logo, a combinação entre a degradação por embaçamento e por ruído gaussiano com SNR=5dB provoca a maior degradação na imagem. Pode-se então dizer que um baixo nível de relação sinal-ruído provoca uma maior concentração de pixels de tons mais claros e, portanto, de uma menor definição e qualidade da imagem em relação à original.

Entre os valores de SNR=10dB e 100dB, a média manteve-se entre 97,39 e 97,41. Uma variação mínima, mas que aponta um ligeiro escurecimento da imagem, com maior predominância de pixels de tons mais escuros em relação à imagem original.

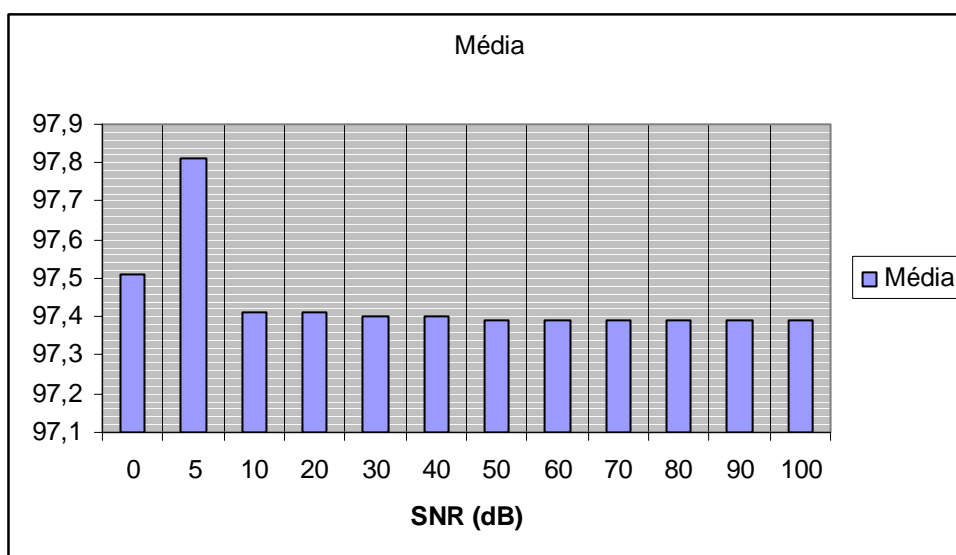


Figura 6.1 – Gráfico – Imagens com degradação – Média.

A Figura 6.2 apresenta os resultados obtidos com a análise dos desvios padrões das amostras trabalhadas.

O desvio padrão é utilizado, pois mede o grau de dispersão dos dados numéricos em torno de um valor médio.

Nota-se que, para a aplicação de degradação por ruído com maiores valores de SNR, o desvio padrão da amostra tende a diminuir em relação ao valor obtido para a imagem original, que foi de 50,94.

A partir de SNR=30dB o valor de desvio padrão tende a manter-se entre 42,79 e 42,80 o que aponta seu distanciamento da imagem original. Ou seja, pode-se verificar que um processo degradante foi efetivamente inserido na imagem.

Ao se observar as imagens no ANEXO B, nota-se que, o resultado visual alia-se ao resultado numérico obtido, já que justamente entre os valores de SNR=30dB e 100dB, é perceptível a homogeneidade citada entre os *pixels*.

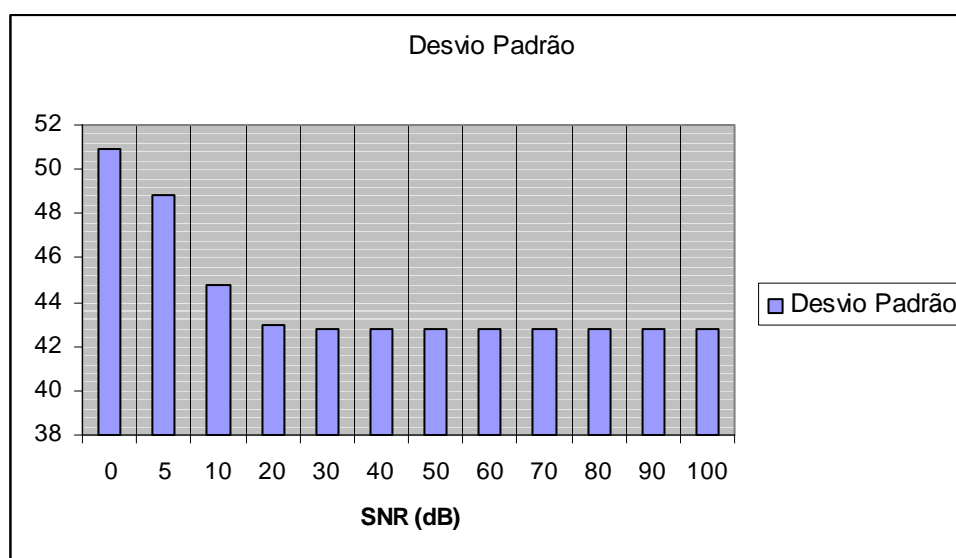


Figura 6.2 – Gráfico – Imagens com degradação – Desvio Padrão.

A inserção de degradação, variando o SNR entre 5dB e 100dB, aponta que a partir de SNR=10dB, a imagem apresenta pixels com tons de cinza mais escurecidos em relação à imagem original, mantendo-se praticamente esta mesma tonalidade até o valor SNR=100dB. E entre SNR=30dB e 100dB, a homogeneidade entre os pixels é numericamente notável.

O número de pixels (538.756) não se altera em nenhuma das imagens.

6.2 - Resultados – Filtragem inversa

A filtragem inversa foi aplicada, como já foi citado, com três valores de “threshold”: 0.00001, 0.05 e 0.1. Na tabela 6.2, estão listados os valores obtidos para cada imagem (original e degradada com SNR=5dB a 100db).

As figuras obtidas com a filtragem inversa e listadas na tabela, podem ser visualizadas no ANEXO C. Com a aplicação do filtro inverso, o número de *pixels* das amostras permaneceu inalterado.

Tabela 6.2 – Resultados: filtragem inversa.

IMAGEM	SNR	TRESHOLD	MÉDIA	DESVIO PADRÃO	ISNR	NÚMERO DE PIXELS
Figura D.1 (Imagem Original)	-	0,000001	106,32	56,12	228,35 dB	538756
Figura D.2 (Imagem Original)	-	0,05	110,29	49,67	5,06 dB	538756
Figura D.3 (Imagem Original)	-	0,1	110,33	47,50	3,38 dB	538756
Figura D.4	5 dB	0,000001	142,38	40,07	-76,66 dB	538756
Figura D.5	5 dB	0,05	126,63	29,68	-11,06 dB	538756
Figura D.6	5 dB	0,1	123,22	34,52	-2,57 dB	538756
Figura D.7	10 dB	0,000001	126,41	47,13	-73,42 dB	538756
Figura D.8	10 dB	0,05	128,85	32,65	-8,22 dB	538756
Figura D.9	10 dB	0,1	121,33	39,86	-0,45 dB	538756
Figura D.10	20 dB	0,000001	125,45	42,61	-63,22 dB	538756
Figura D.11	20 dB	0,05	110,47	39,78	-0,84 dB	538756
Figura D.12	20 dB	0,1	113,68	46,73	2,59 dB	538756
Figura D.13	30 dB	0,000001	141,20	40,85	-56,67 dB	538756
Figura D.14	30 dB	0,05	110,96	47,70	3,94 dB	538756
Figura D.15	30 dB	0,1	112,33	47,26	3,29 dB	538756
Figura D.16	40 dB	0,000001	133,96	46,34	-44,15 dB	538756
Figura D.17	40 dB	0,05	110,98	49,96	4,93 dB	538756
Figura D.18	40 dB	0,1	110,37	47,98	0,1 dB	538756
Figura D.19	50 dB	0,000001	121,52	33,60	-31,12 dB	538756
Figura D.20	50 dB	0,05	110,06	49,65	5,05 dB	538756
Figura D.21	50 dB	0,1	110,50	47,51	3,37 dB	538756
Figura D.22	60 dB	0,000001	134,07	41,27	-24,34 dB	538756
Figura D.23	60 dB	0,05	110,22	49,65	5,06 dB	538756
Figura D.24	60 dB	0,1	110,39	47,51	3,38 dB	538756
Figura D.25	70 dB	0,000001	118,50	34,30	-10,45 dB	538756
Figura D.26	70 dB	0,05	110,31	49,69	5,06 dB	538756
Figura D.27	70 dB	0,1	110,33	47,49	3,38 dB	538756
Figura D.28	80 dB	0,000001	120,32	41,31	-0,42 dB	538756
Figura D.29	80 dB	0,05	110,32	49,67	5,06 dB	538756
Figura D.30	80 dB	0,1	110,33	47,51	3,38 dB	538756
Figura D.31	90 dB	0,000001	111,69	50,06	7,29 dB	538756
Figura D.32	90 dB	0,05	110,29	49,67	5,06 dB	538756
Figura D.33	90 dB	0,1	110,33	47,51	3,38 dB	538756
Figura D.34	100 dB	0,000001	106,29	54,41	17,82 dB	538756
Figura D.35	100 dB	0,05	110,30	49,67	5,06 dB	538756
Figura D.36	100 dB	0,1	110,33	47,50	3,38 dB	538756

A partir dos resultados listados na Tabela 6.2, foram gerados gráficos para média, desvio padrão e ISNR.

6.2.1 - Média

A Figura 6.3 apresenta o gráfico dos valores encontrados para a média dos valores dos pixels. Para facilitar a comparação com a imagem original, os valores obtidos para a mesma estão dispostos na coluna de valor SNR=0dB.

Iniciando-se a análise gráfica pelos valores obtidos para “*threshold*”=0,00001 - representados no gráfico pela cor roxa - verifica-se que tais valores relatam uma grande variação entre as amostras. Tem-se com isso o fato de que para este valor de “*threshold*”, a médias dos pixels das imagens entre SNR=5dB e 60dB apresenta predominância de colorações mais claras de *pixels* (valores mais elevados), se comparadas à imagem original. Neste intervalo, a verificação visível da restauração não se torna possível (ver ANEXO C). De um modo geral, portanto, a aplicação da restauração por filtragem inversa com valor de “*threshold*”=0,000001 não obteve resultados satisfatórios por ocasionar perda de informações visuais da imagem.

Os valores obtidos com “*threshold*”=0,05 e 0,1, apresentaram resultados bastante próximos. No gráfico da Figura 6.3, foram adotadas para “*threshold*”=0,05 e “*threshold*”=0,1, as colorações vermelha e amarela, respectivamente. Observa-se no gráfico da Figura 6.3, que seus valores permanecem praticamente igualados aos valores obtidos para a imagem original. Sua aplicação no processo de restauração manteve o valor médio dos pixels da amostra original, o que acarretou a obtenção de resultados compatíveis com o original.

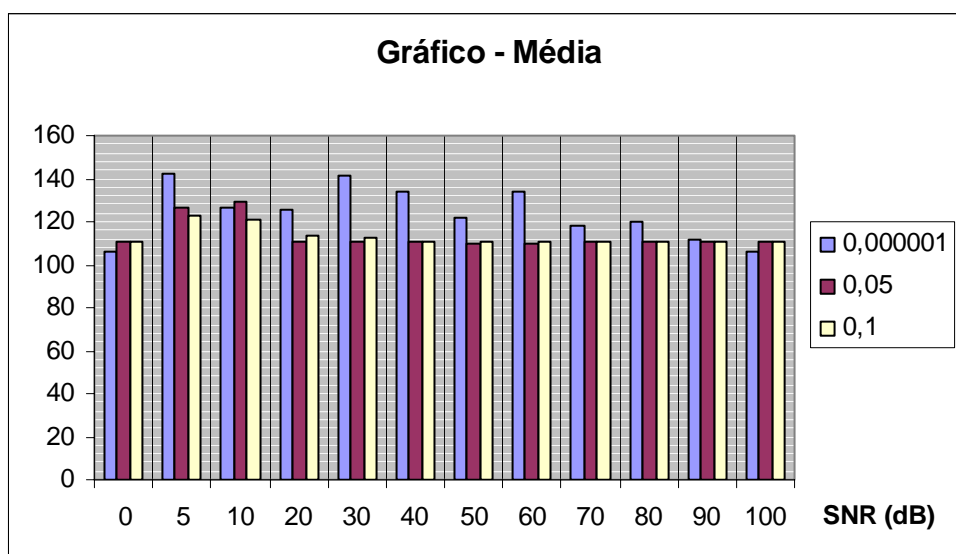


Figura 6.3 – Gráfico – Filtragem inversa – Média.

6.2.2 – Desvio Padrão

Para resultados mais homogêneos e uniformes, o desvio padrão deve-se manter próximo ao da imagem original, como apresenta o gráfico da Figura 6.4.

Os melhores resultados foram obtidos para “*threshold*” = 0,05 e 0,1, na faixa entre 30dB e 100dB.

Novamente, os valores para “*threshold*”=0,000001 não foram satisfatórios.

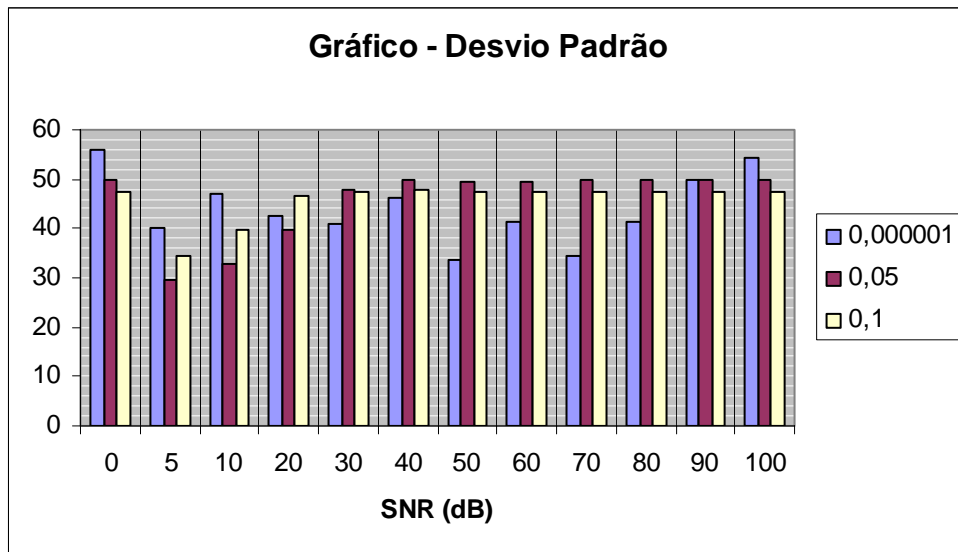


Figura 6.4 – Gráfico – Filtragem inversa – Desvio Padrão.

6.2.3 - ISNR

O ISNR não necessita ter um valor próximo ao encontrado para a imagem original, já que mede a melhoria da imagem a partir do índice de SNR aplicado à mesma.

Com o resultado obtido para a INSR, pode-se dizer qual foi o melhor resultado e o pior resultado da aplicação da filtragem. Essa relação é compatível numericamente e visualmente.

Os valores obtidos para o ISNR, estão graficamente dispostos na Figura 6.5. Assim como obtido para a média e o desvio padrão, os valores de “*threshold*”=0,05 e 0,1 tendem a manter-se praticamente iguais no intervalo entre SNR=30dB e 100dB.

A aplicação do valor de “*threshold*”=0,000001 obteve o melhor resultado da restauração com SNR=100dB. Ou seja, o melhor resultado da restauração inversa foi obtido com o menor valor utilizado para “*threshold*” e o maior valor de SNR.

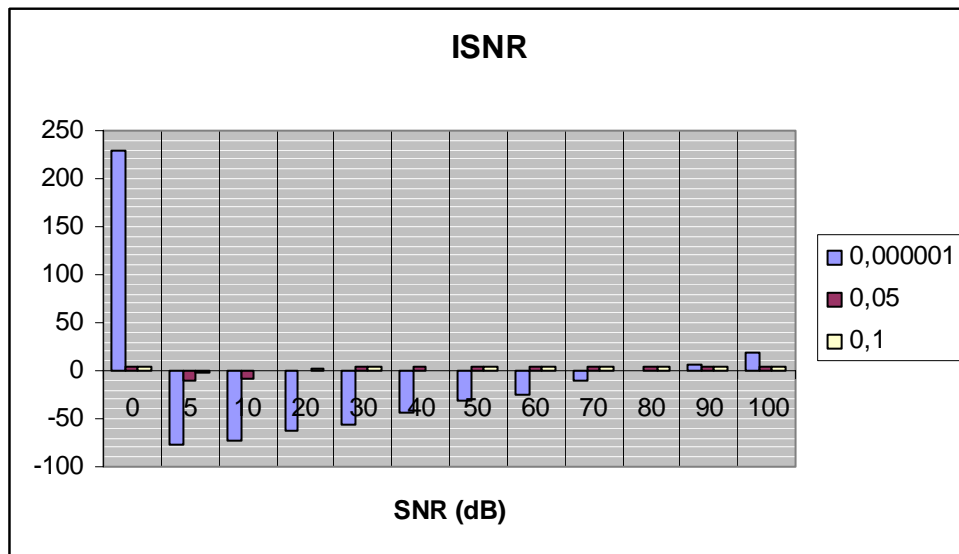


Figura 6.5 – Gráfico – Filtragem inversa – ISNR.

6.2.4 – Análise Geral

Com a aplicação da filtragem inversa, concluiu-se que:

- Quantitativamente, os melhores resultados foram obtidos utilizando-se os parâmetros de “*threshold*” iguais a 0.05 e 0.1;
- Qualitativamente, a média e o desvio padrão devem manter-se com valores pouco variáveis em relação a tais valores da imagem original;
- O melhor resultado da restauração inversa foi obtido com o menor valor utilizado para “*threshold*” (0,000001) e o maior valor de SNR (100dB), com ISNR = 17,82 dB.
- O pior resultado da restauração inversa foi obtido com o menor valor utilizado para “*threshold*” (0,000001) e o menor valor de SNR (5dB), com ISNR = -76,66 dB.
- A relação entre o melhor resultado e o pior resultado obtido, mostra que o valor próximo à origem para a função $H[u,v]$, realmente obtém melhores resultados quando existe um alto valor de SNR, ou seja, uma boa relação sinal-ruído;
- Além disso, ao observar o gráfico da aplicação do SNR nas Figuras 6.1 e 6.2, observa-se que, entre SNR=30dB e SNR=100dB os valores de média e desvio padrão tendem a permanecer com valores muito próximos. Aplicando-se a

filtragem inversa, nota-se que com “*threshold*”=0,05 e 0,1, os resultados tendem a apresentar uma homogeneidade entre essa mesma faixa de valores.

6.3 - Resultados – Filtragem Wiener

A filtragem Wiener foi aplicada nas mesmas amostras utilizadas na filtragem inversa (original e degradadas com SNR variando de 5dB a 100db).

As figuras obtidas com a filtragem Wiener e listadas na Tabela 6.3, podem ser visualizadas no ANEXO D. Com a aplicação do filtro Wiener, o número de *pixels* das amostras permaneceu inalterado.

Tabela 6.3 – Resultados: filtragem Wiener.

IMAGEM	SNR	MÉDIA	DESVIO PADRÃO	ISNR	NÚMERO DE PIXELS
Figura E.1 (Imagem Original)	-	160,32	56,12	228,35 dB	538756
Figura E.2	5 dB	110,11	49,44	5,07 dB	538756
Figura E.3	10 dB	114,19	49,55	3,83 dB	538756
Figura E.4	20 dB	108,36	49,51	4,14 dB	538756
Figura E.5	30 dB	113,88	49,46	6,04 dB	538756
Figura E.6	40 dB	107,83	51,67	8,42 dB	538756
Figura E.7	50 dB	108,21	53,75	11,38 dB	538756
Figura E.8	60 dB	107,43	52,95	14,72 dB	538756
Figura E.9	70 dB	108,42	54,52	18,47 dB	538756
Figura E.10	80 dB	107,47	55,30	22,93 dB	538756
Figura E.11	90 dB	106,43	55,75	27,72 dB	538756
Figura E.12	100 dB	106,49	56,00	32,98 dB	538756

6.3.1 – Média, Desvio Padrão e ISNR

Para a filtragem Wiener, foram plotados em um mesmo gráfico, todos os valores de média, desvio padrão e ISNR encontrados e são representados pelas cores roxo, vermelho e amarelo, respectivamente.

A filtragem Wiener apresentou um aspecto satisfatório em seus resultados.

Observando o comportamento dos valores do desvio padrão, verifica-se que se mantiveram muito próximos aos da imagem original, para todos os valores de SNR aplicados, variando entre 49,44 e 56,12. Isso explica a homogeneidade nas imagens obtidas.

A média e o ISNR apresentaram um crescimento gradativo. Esse crescimento é diretamente proporcional à melhoria visual da imagem (ver ANEXO D). Ou seja, para maiores valores de média, foram encontrados os melhores resultados.

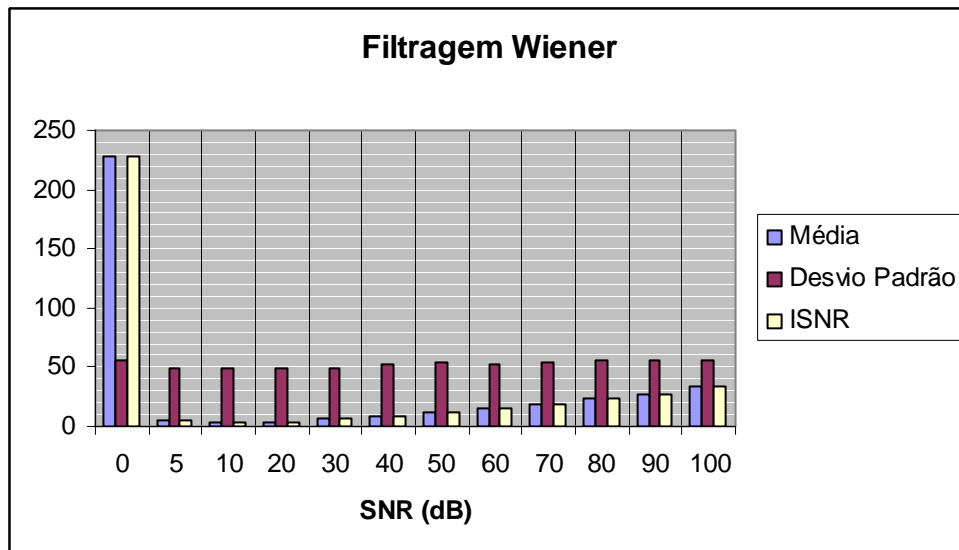


Figura 6.6 – Gráfico – Filtragem Wiener – Média, Desvio Padrão e ISNR.

6.3.2 – Análise Geral

Com a aplicação da filtragem Wiener, concluiu-se que:

- Os resultados obtidos apresentam maior consistência, já que são evidentemente crescentes de acordo com a melhoria na relação sinal-ruído aplicadas;
- O desvio padrão manteve-se praticamente constante para todas as amostras, o que explica a homogeneidade das imagens em relação à original;
- A média e o ISNR tiveram um crescimento gradativo e compatível com a melhoria visual das imagens;
- O melhor resultado foi obtido para SNR=100dB, com ISNR=32,98dB.
- O pior resultado foi obtido para SNR=5dB, com ISNR=5,07dB.
- A filtragem Wiener teve um resultado satisfatório e eficiente, comprovando o fato de ser mais eficiente que a filtragem inversa para a restauração de imagens degradadas com ruído.

6.4 – Resultados - Conclusão

Para o caso da filtragem inversa, utilizando-se o parâmetro “*threshold*” para delimitar $H[u,v]$ próximo à origem, verifica-se que para uma boa relação sinal-ruído (100dB) realmente é uma técnica eficaz, para um valor pequeno de “*threshold*”, como 0,000001. Para esses valores, foi obtido o melhor resultado da simulação, com ISNR = 17,82 dB.

Foram utilizados também os parâmetros de “*threshold*” = 0,05 e 0,1. Tais parâmetros apresentaram resultados satisfatórios e com uma variação mínima, na faixa que vai de SNR = 30dB e 100dB. Nesta mesma faixa de valores, os resultados de média e desvio padrão apontados para as imagens durante a inserção de degradação por ruído, também se mantiveram praticamente constantes.

Em termos de quantidade, o maior número de resultados com características visuais compreensíveis, foi obtido com valores de “*threshold*” = 0,05 e 0,1.

Em termos de qualidade, o melhor resultado foi obtido com o menor valor de “*threshold*” = 0,000001 e o maior valor de SNR (100dB). Além disso, pode-se dizer também que a qualidade da imagem depende da média e do desvio padrão se manterem com valores próximos aos da imagem original.

A filtragem Wiener apresentou um desempenho excelente, com uma melhoria claramente perceptível numericamente e visualmente. Seu comportamento descreve-se por: manter o desvio padrão praticamente constante (com valor próximo ao da imagem original), média e ISNR gradativamente crescentes de acordo com o aumento da relação sinal-ruído.

A Figura 6.7 mostra os melhores resultados obtidos com cada tipo de filtragem. O melhor resultado obtido com a filtragem Wiener foi com SNR=100dB, apresentando ISNR = 32,98 dB. Ou seja, o melhor índice de melhoria da imagem obtido com a filtragem Wiener foi superior ao obtido com a filtragem inversa (17,82 dB).

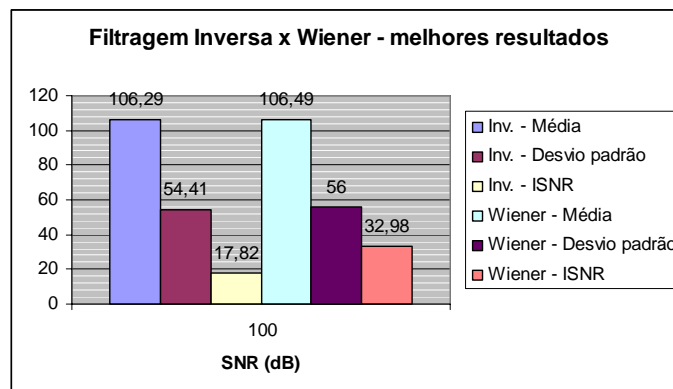


Figura 6.7 – Gráfico - Filtragem inversa x Wiener – melhores resultados.

Para os melhores resultados obtidos, a média e desvio padrão não apresentaram variações bruscas em seus valores entre os dois métodos. Mas o ISNR demonstra ser um fator importante na análise comparativa, já que de fato aponta a imagem obtida com melhor índice de restauração.

A restauração com filtro Wiener demonstrou ser o método mais eficiente e eficaz na restauração de imagens degradadas por embaçamento e ruído gaussiano, apresentando resultados superiormente satisfatórios em comparação à filtragem inversa.

6.5 – Melhores resultados obtidos

6.5.1 – Filtragem inversa

Melhor resultado obtido para a filtragem inversa.

O melhor resultado foi obtido com SNR=100dB, thresh=0,000001dB, apresentando ISNR=17,82dB.

Rest Inv Img SNR=100dB, thresh=0.000001, ISNR = 17.82 dB



Figura 6.8 – Filtragem inversa – melhor resultado.

6.5.2 – Filtragem Wiener

Melhor resultado obtido para a filtragem inversa.

O melhor resultado foi obtido com SNR=100dB, ISNR=32,98dB.

Rest Wiener Img SNR=100dB, ISNR = 32.98 dB



Figura 6.8 – Filtragem Wiener – melhor resultado.

Capítulo 7 - Conclusão

7.1 – Considerações finais

Neste trabalho foram apresentadas duas técnicas de restauração de imagens digitais. Fez-se uma breve apresentação sobre a imagem digital e da base teórica necessária para o entendimento necessário do assunto abordado.

Foi apresentado o processo que ocorre matematicamente para a aplicação de um filtro de restauração em uma imagem, com seus modelos de inserção de degradação e de restauração.

Das duas técnicas utilizadas nas simulações, a filtragem Wiener apresentou os melhores resultados, principalmente para maiores valores de relação sinal-ruído.

A filtragem inversa apresentou ser eficaz na recuperação de imagens degradadas, mas seu melhor resultado de ISNR foi menor que o obtido com a filtragem Wiener.

As duas técnicas possuem detalhes importantes como: para a implementação do filtro Wiener é necessário estimar o espectro de potência da imagem original e do ruído, o que é facilmente obtido com o uso de algoritmos adequados. Para a implementação do filtro inverso, é necessário definir um valor de $H[u,v]$ próximo à origem do espectro de frequências para que o ruído não predomine sobre a imagem com a aplicação da filtragem.

A partir dos resultados obtidos, chega-se à conclusão de que o filtro Wiener é mais eficaz nos casos de degradação com presença de ruído aditivo. Além disso, a filtragem Wiener possui um tratamento matemático mais robusto, já que possui tratamento de erro e tem ainda um comportamento numericamente confiável, que pôde ser observado no gráfico plotado a partir de seus dados tabelados.

Os dois métodos utilizados foram de simples implementação, com algoritmos leves (na utilização da imagem com as dimensões de 256×256 *pixels*) e que, para a imagem utilizada, não dependeram de esforço computacional exagerado.

Contribuindo com as pesquisas e estudos científicos e acadêmicos, este trabalho possui a base matemática dos métodos de restauração utilizados. Possui ainda, resultados tabelados que demonstram o comportamento de ambos os filtros, promovendo um auxílio a futuras comparações com outros métodos e com a implementação de softwares que realizem tal tipo de processamento de imagens. Contribuindo com a tecnologia, softwares podem ser

desenvolvidos de modo a otimizar o processo de restauração promovendo um ambiente visual que possibilite sua utilização por usuários clientes.

A restauração de imagens é uma prática bastante utilizada por grandes corporações que necessitam de tratamento de imagens. Por apresentar o código utilizado disponível à visualização, a aplicação utilizada neste trabalho pode auxiliar que futuros estudos possam ser desenvolvidos visando usuários comuns (domésticos), pequenas corporações e o tratamento de imagens em geral.

7.2 – Trabalhos Futuros

A restauração de imagens é um campo amplo, com inúmeras possibilidades de atuação.

Existem vários softwares no mercado que apenas realçam imagens, melhorando brilho, contraste e características visuais, mas que utilizam procedimentos subjetivos de melhoria da imagem, fazendo uso de características visuais e cognitivas da visão humana.

Os softwares de restauração de imagens existentes atualmente no mercado, são produzidos pelas próprias corporações que fazem tratamento de imagens, de modo que sejam personalizados e atendam às suas necessidades.

Seria um grande avanço no campo de PDI, o desenvolvimento de um software doméstico de restauração de imagens, que não necessite de grandes e potentes equipamentos para obter bom desempenho. Com isso, empresas, universidades, centros de pesquisas em imagens podem obter uma ferramenta de auxílio na remoção de degradações em imagens das quais fazem uso.

Além disso, o algoritmo utilizado na simulação deste trabalho, pode ser melhorado de forma a associar novos métodos avaliativos de desempenho, novas técnicas de filtragem, novas formas de se obter os resultados e o tratamento de imagens coloridas e testes com maiores proporções de imagens. E ainda, a partir do código fonte utilizado podem-se fazer aplicações mais robustas em outras linguagens de programação que não dependam dos recursos oferecidos pelo software Matlab.

Referências Bibliográficas

AHUMADA, Albert. **Classification Image and Noise Estimation - NASA Ames Research Center, Moffett Field, CA, USA.** Disponível em: <<http://vision.arc.nasa.gov/personnel/al/papers/02jov/ClassIm.htm>> . Acesso em: 18 de outubro de 2004.

BERROCOSO, Jesús V. **Elementos morfológicos: el color.** 2003. Disponível em: <http://www.unex.es/didactica/Tecnologia_Educativa/imagen10.htm>. Acesso em: 11 de janeiro de 2005.

BOURKE, Paul. **Image Filtering in the Frequency Domain.** Disponível em: <<http://astronomy.swin.edu.au/~pbourke/analysis/imagefilter/>>. Acesso em: 12 de janeiro de 2005.

BRAVIANO, Gilson. **Tratamento de Imagens.** Universidade Federal de Santa Catarina – UFSC. Departamento de Expressão Gráfica – EGR.

CARVALHO, Marco A. **Processamento Digital de Imagens – ST061.** Campinas: Material didático de curso superior, 2004. Disponível em: <<http://www.ceset.unicamp.br/webdidat/matdidat.php?cod=ST061&nome=Marco+Antonio+Garcia+de+Carvalho>>. Acesso em: 01 de setembro de 2004.

CATANZARO, Julian. **Image Restoration.** Disponível em: <<http://www.julian.catanzaro.com/science/517imagerestoration.pdf>>. Acesso em: 09 de novembro de 2004.

DANGETI, Sarita Veera. **Denoising Techniques – A Comparison.** Disponível em: <<http://etd02.lnx390.lsu.edu/docs/available/etd-1219102152426/unrestricted/thesisreportfinal.pdf>>. Acesso em: 13 de novembro de 2004.

DE BIASI, Herculano. **Técnicas de Filtragem.** Disponível em: <<http://www.inf.ufsc.br/~visao/1998/herculan>>. Acesso em: 03 de novembro de 2004.

Departamento de Física da Universidade Federal do Ceará. **As Séries de Fourier.** Disponível em: <<http://www.fisica.ufc.br/fourier/fourier2.htm>>. Acesso em: 10 de setembro de 2004.

FERREIRA, Vital. **Restauração de Imagens Digitais.** Disponível em: <http://www.sel.eesc.sc.usp.br/homero/aulas/sel708/seminarios/restauracao_arquivos/frame.htm#slide0016.htm>. Acesso em: 15 de outubro de 2004.

FISHER, Robert. **Image Transformations and Filters.** Disponível em: <<http://homepages.inf.ed.ac.uk/rbf/CVonline/transf.htm>>. Acesso em: 21 de outubro de 2004.

FISHER, Bob; PERKINS, Simon; WALKER, Ashley; WOLFART, Erik. **Hypermedia Image Processing Reference.** 2003. Disponível em: <<http://www.cee.hw.ac.uk/hipr/html/histogram.html>>. Acesso em: 18 de novembro de 2004.

FRIGO, Matteo; JOHNSON, Steven G. **FFTW.** 2004. Disponível em <<http://www.fftw.org/fftw3.pdf>>. Acesso em: 11 de janeiro de 2005.

GLOSSÁRIO UOL. Disponível em: <<http://www1.uol.com.br/publicidade/glossario.htm#E>>. Acesso em: 11 de janeiro de 2005.

GONZALEZ, Rafael C.; WOODS, Richard E. **Digital Image Processing.** Editora Pearson Education Asia. Segunda Edição. 2002.

GRAHAM, James. **Theory of the DFT.** Disponível em: <<http://astron.berkeley.edu/~jrg/ngst/fft/dftideas.html>>. Acesso em: 09 setembro de 2004.

HAETINGER, Claus. **Ferramentas de Análise: Aplicações da FFT.** Disponível em: <http://ensino.univates.br/~chaet/Materiais/Cap%edotulo_12_Bloch.pdf>. Acesso em: 10 de novembro de 2004.

HARE, James; REILLY, James. **The Iterative Deconvolution Of Linearly Blurred Images Using Non-Parametric Stabilizing Functions.** 2004. Disponível em: <<http://www.ece.mcmaster.ca/~reilly/html/papers/reilly/1988.pdf>>. Acesso em: 27 de novembro de 2004.

HILLEN, Walter. **Frequency Domain & Image Restoration.** Disponível em: <http://www.medizinischeinformatik.fh-aachen.de/downloads/master/6_Img_Restor_monitor.pdf>. Acesso em: 01 de novembro de 2004.

HIRATA, Nina. **Restauração de Imagens.** Disponível em: <<http://www.vision.ime.usp.br/~nina/cursos/mac0417-02/aula14b.html>>. Acesso em: 14 de outubro de 2004.

HOUAISS – Dicionário Eletrônico da Língua Portuguesa. Versão 1.0.5. Editora Objetiva Ltda. Agosto de 2002.

JÄHNE, Bernd. **Digital Image Processing.** Germany: Springer. 2002

JIANG, Ming. **Wiener Filtration.** 2003. Disponível em: <<http://ct.radiology.uiowa.edu/~jiangm/courses/dip/html/node117.html>>. Acesso em: 13 de outubro de 2004.

JIANG, Ming. **Digital Image Processing.** Disponível em: <<http://ct.radiology.uiowa.edu/~jiangm/courses/dip/html/node106.html>>. Acesso em: 03 de outubro de 2004.

JONKER, Pieter. **Thresolding.** Disponível em : <<http://www.ph.tn.tudelft.nl/Courses/FIP/frames/fip-Segmenta.html>>. Acesso em: 17 de novembro de 2004.

JPEG FAQ. Disponível em: < <http://www.faqs.org/faqs/jpeg-faq/>>. 2004. Acesso em: 11 de janeiro de 2005.

KAISER, Peter. **The Joy of Visual Perception – A Web Book.** Disponível em: <<http://www.yorku.ca/eye/thejoy.htm>>. Acesso em: 16 de setembro de 2004.

KATSAGGELOS, Aggelos K; BANHAM, Mark R. **Digital Image Restoration.** IEEE SIGNAL PROCESSING MAGAZINE. Março, 1997.

KINAPE, Rogério M.; AMORIM, Mardson F. **Medidas computacionais em qualidade de imagem: uma comparação.** Pontifícia Universidade Católica do Paraná (PUC/PR).

KOKARAM, Anil. **LENNA.jpg**. 2003 Disponível em: <<http://www.mee.tcd.ie/~ack/cd/linereg/lenna.jpg>>. Acesso em: 12 de janeiro de 2005.

MACHADO, Alexei M. C. **Processamento Digital de Imagens**. Disponível em: <<http://www.inf.pucmg.br/projetos/latim/estudobiblio/resumopi.htm>>. Pontifícia Universidade Católica de Minas - PUC Minas. Acesso em: 02 de março de 2005.

MAI, Luong. **Introduction to Computer Vision and Image Processing**. Disponível em: <<http://www.netnam.vn/unescocourse/computervision/91.htm>>. Acesso em: 19 de outubro de 2004.

MAOR, Eli. **Trigonometric Delights**. New Jersey: Priceton University Press, 1998.

MATHWORKS. Disponível em: <<http://www.mathworks.com/products/matlab/>>. Acesso em: 04 de janeiro de 2004.

NETO, João F. **Aplicação de Fourier no Processamento Digital de Imagens**. Disponível em: <<http://www.infonet.com.br/users/jfonseca/PDI-FOURIER.PDF>>. Acesso em: 21 de novembro de 2004.

NOCETI, Sidnei. **Fundamentos Sobre Ruídos**. Disponível em: <<http://www.linse.ufsc.br/~sidnei/index2.htm>>. Acesso em: 28 de outubro de 2004.

OLABARRIAGA, Sílvia. **Restauração de Imagens**. Disponível em: <<http://www.inf.ufrgs.br/~silvia/ipg/slidesRestaura.pdf>>. Acesso em: 25 de outubro de 2004.

PACIORNIK, Sidney. **A Transformada de Fourier Discreta**. Disponível em: <<http://www.dcm.puc-rio.br/cursos/ipdi/html/sld078.htm>>. Acesso em: 09 de setembro de 2004.

RINTOUL, Jean. **Theory of FFT Wiener Filtering**. 2000. Disponível em: <<http://oldsite.vislab.usyd.edu.au/education/sc3/2000/jean/wiener/fft.html>>. Acesso em: 13 de novembro de 2004.

SCHWARTZ, William R. **Transformadas de Imagens: Teoria e Aplicações**. Disponível em: <<http://pet.inf.ufpr.br/~william/papers/tg.pdf>>. Acesso em: 27 de novembro de 2004.

SCURI, Antonio E. **Fundamentos da Imagem Digital**. Tecgraf/PUC-Rio. Agosto 2002.

SHIMIZU, K. **Telemedicine by mobile communication**. Artigo IEEE. 1999.

SILVA, Ana Paula C.; SANTOS, Evelize P.; LOURENÇÃO, Neyre C. M. **Glossário de Processamento de Imagens, 1ª Edição**. Disponível em <http://www.unibero.edu.br/download/glossarios/glossario_processamentodeimagens.doc>. Acesso em: 21 de novembro de 2004.

SMITH, Steven W. **Digital Signal Processing – A practical guide for Engineers and Scientists**. EUA: Ed. Newnes. 2003.

SMITH, Steven W. **The Scientist and Engineer's Guide to Digital Signal Processing**. Disponível em: <<http://www.dspguide.com/ch28.htm>>. Acesso em: 27 de novembro de 2004.

SOBRAL, João. **Visão por Computador**. Disponível em: <<http://gec.di.uminho.pt/lesi/vpc0304>>. Acesso em: 19 de outubro de 2004.

SUTTON, Eric. **Image Processing & Computer Vision: Handouts**. Disponível em: <<http://www.ecsutton.ece.ufl.edu/dip/handouts>>. Acesso em: 24 de setembro de 2004.

The Encyclopaedia of Medical Imaging Volume I. **Point-spread Function (PSF)**. Disponível em: <<http://www.amershamhealth.com/medcyclopaedia/medical/volume%20I/POINT%20SPREAD%20FUNCTION%20%20PSF%20.asp>>. Acesso em: 15 de outubro de 2004.

THESSALONIKI. **Dip_material**. Disponível em: <http://poseidon.csd.auth.gr/LAB_PUBLICATIONS/Books/dip_material/>. Acesso em: 09 de novembro de 2004.

TOLEDO, Rodrigo. **Matemática por trás de Processamento de Imagens**. Disponível em: <[http://www.tecgraf.puc-rio.br/~rtoledo/cg1/ICG_04%20\(proc%20imagem\).ppt#2](http://www.tecgraf.puc-rio.br/~rtoledo/cg1/ICG_04%20(proc%20imagem).ppt#2)>. Acesso em: 02 de novembro de 2004.

VELDHUIZEN, Todd. **The Image Restoration Problem**. Disponível em: <http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/VELDHUIZEN/node5.html>. Acesso em: 18 de outubro de 2004.

VELHO, Luiz; TEIXEIRA, Ralph. **Processamento de Imagens**. 2000. Disponível em: <http://www.visgraf.impa.br/Courses/ip00/proj/Deconv/processamento_de_imagens.htm>. Acesso em: 25 de outubro de 2004.

WANDERLEY, Juliana. **Aulas de Processamento de Imagens**. Disponível em: <<http://www.ene.unb.br/~juliana/cursos/pimagens/aulas.html>>. Acesso em: 19 de outubro de 2004.

WANG, Cherry; ZHANG, Huipin; DORAN, Andrew. **Image Restoration**. Disponível em: <<http://www.owl.net.rice.edu/~elec539/Projects99/BACH/proj2/>>. 1999. Acesso em: 06 de outubro de 2004.

WEN, Che-Yen; LEE, Chien-Hsiung. **Point spread functions and their applications to forensic image restoration**. Forensic Science Journal. 2002. Disponível em: <[http://sun4.cpu.edu.tw/~fsjournal/content/vol1.no.1/p2\(15-26\).pdf](http://sun4.cpu.edu.tw/~fsjournal/content/vol1.no.1/p2(15-26).pdf)>. Acesso em: 15 de novembro de 2004.

WHOSSACK, Will. **Digital Image Analysis and Theory of Image Processing**. Department of Physics & Astronomy of University of Edinburgh. Disponível em: <<http://www.ph.ed.ac.uk/~wjh/teaching/dia/index.html>>. Acesso em: 13 de novembro de 2004.

YAMASHITA, Yukihiro. **Optimum Sampling Vectors for Wiener Filter Noise Reduction**. Disponível em: <<http://www.ee.udel.edu/~barner/journalmeeting/Papers/00972482.pdf>>. Acesso em: 09 de novembro de 2004.

ANEXO A – Código Fonte em Matlab

Observações:

- Os comentários são iniciados pelo símbolo de porcentagem (%).

%Abrindo a imagem Lenna.jpg

```
f = double(imread('C:\MATLAB6p5\lenna.jpg'))/255;  
f = f(10+[1:256],10+[1:256],:);  
figure(1);imshow(f,[0 1]),title('Imagem Original')
```

%Degradação por embaçamento – definir matriz

```
hc = zeros(256);  
hc([1:6 252:256 1:6 252:256],[1:6 1:6 252:256 252:256]) = 1/(11^2);
```

%Aplicação de embaçamento

```
g = real(ifft2( fft2(f).*fft2(hc) ));  
figure(2);imshow(g,[0 1]),title('Imagem Original Embaçada')
```

%Inserção de embaçamento ruído gaussiano na imagem embaçada, com SNR variando entre 5dB e 100dB

```
sigmaG = std2(g);
```

%Imagem Degradada com SNR=5db

```
sigma5 = sqrt( sigmaG^2 / 10^(5/10) );  
g5 = imnoise(g,'gaussian',0,(sigma5)^2);  
figure(3);imshow(g5,[0 1]),title('Imagem degradada com SNR = 5dB')
```

%Imagem Degradada com SNR =10dB

```
sigma10 = sqrt( sigmaG^2 / 10^(10/10) );  
g10 = imnoise(g,'gaussian',0,(sigma10)^2);  
figure(4);imshow(g10,[0 1]),title('Imagem degradada com SNR = 10dB')
```

%Imagem Degradada com SNR =20dB

```
sigma20 = sqrt( sigmaG^2 / 10^(20/10) );  
g20 = imnoise(g,'gaussian',0,(sigma20)^2);  
figure(5);imshow(g20,[0 1]),title('Imagem degradada com SNR = 20dB')
```

%Imagem Degradada com SNR =30dB

```
sigma30 = sqrt( sigmaG^2 / 10^(30/10) );  
g30 = imnoise(g,'gaussian',0,(sigma30)^2);  
figure(6);imshow(g30,[0 1]),title('Imagem degradada com SNR = 30dB')
```

%Imagem Degradada com SNR =40dB

```
sigma40 = sqrt( sigmaG^2 / 10^(40/10) );  
g40 = imnoise(g,'gaussian',0,(sigma40)^2);  
figure(7);imshow(g40,[0 1]),title('Imagem degradada com SNR = 40dB')
```

%Imagem Degradada com SNR =50dB

```
sigma50 = sqrt( sigmaG^2 / 10^(50/10) );  
g50 = imnoise(g,'gaussian',0,(sigma50)^2);  
figure(8);imshow(g50,[0 1]),title('Imagem degradada com SNR = 50dB')
```

%Imagem Degradada com SNR =60dB

```
sigma60 = sqrt( sigmaG^2 / 10^(60/10) );  
g60 = imnoise(g,'gaussian',0,(sigma60)^2);  
figure(9);imshow(g60,[0 1]),title('Imagem degradada com SNR = 60dB')
```

%Imagem Degradada com SNR =70dB

```
sigma70 = sqrt( sigmaG^2 / 10^(70/10) );  
g70 = imnoise(g,'gaussian',0,(sigma70)^2);  
figure(10);imshow(g70,[0 1]),title('Imagem degradada com SNR = 70dB')
```

%Imagem Degradada com SNR =80dB

```
sigma80 = sqrt( sigmaG^2 / 10^(80/10) );  
g80 = imnoise(g,'gaussian',0,(sigma80)^2);  
figure(11);imshow(g80,[0 1]),title('Imagem degradada com SNR = 80dB')
```

%Imagem Degradada com SNR =90dB

```
sigma90 = sqrt( sigmaG^2 / 10^(90/10) );  
g90 = imnoise(g,'gaussian',0,(sigma90)^2);  
figure(12);imshow(g90,[0 1]),title('Imagem degradada com SNR = 90dB')
```

%Imagem Degradada com SNR =100dB

```
sigma100 = sqrt( sigmaG^2 / 10^(100/10) );  
g100 = imnoise(g,'gaussian',0,(sigma100)^2);  
figure(13);imshow(g100,[0 1]),title('Imagem degradada com SNR = 100dB')
```

%FILTRO INVERSO

```
H = fft2(hc);
```

%Imagem Original

```
thresh = 0.000001;  
bTh = abs(H)<thresh;  
G = fft2(g);  
FR = G./H;  
FR(bTh) = 0;  
fr = real(ifft2(FR));  
ISNR = 10*log10( sum([f(:)-g(:)].^2)/sum([f(:)-fr(:)].^2) );  
figure(14);imshow(fr,[]);title(sprintf('Rest Inv Img Original, thresh=0.000001, ISNR = %.2f  
dB',ISNR))
```

```
thresh = 0.05;  
bTh = abs(H)<thresh;  
G = fft2(g);  
FR = G./H;  
FR(bTh) = 0;
```

```

fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(15);imshow(fr,[]);title(sprintf('Rest Inv Img Original, thresh=0.05, ISNR = %.2f
dB',ISNR))

```

```

thresh = 0.1;
bTh = abs(H)<thresh;
G = fft2(g);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(16);imshow(fr,[]);title(sprintf('Rest Inv Img Original, thresh=0.1, ISNR = %.2f
dB',ISNR))

```

% Imagem SNR=5db

```

thresh = 0.000001;
bTh = abs(H)<thresh;
G = fft2(g5);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g5(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(17);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=5dB, thresh=0.000001, ISNR =
%.2f dB',ISNR))

```

```

thresh = 0.05;
bTh = abs(H)<thresh;
G = fft2(g5);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g5(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(18);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=5dB, thresh=0.05, ISNR = %.2f
dB',ISNR))

```

```

thresh = 0.1;
bTh = abs(H)<thresh;
G = fft2(g5);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g5(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(19);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=5dB, thresh=0.05, ISNR = %.2f
dB',ISNR))

```

%Imagem SNR=10db

```

thresh = 0.000001;
bTh = abs(H)<thresh;

```



```

G = fft2(g10);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g10(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(20);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=10dB, thresh=0,00001, ISNR = %.2f dB',ISNR))

```

```

thresh = 0.05;
bTh = abs(H)<thresh;
G = fft2(g10);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g10(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(21);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=10dB, thresh=0.05, ISNR = %.2f dB',ISNR))

```

```

thresh = 0.1;
bTh = abs(H)<thresh;
G = fft2(g10);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g10(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(22);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=10dB, thresh=0.05, ISNR = %.2f dB',ISNR))

```

%Imagem SNR=20db

```

thresh = 0.000001;
bTh = abs(H)<thresh;
G = fft2(g20);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g20(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(23);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=20dB, thresh=0.00001, ISNR = %.2f dB',ISNR))

```

```

thresh = 0.05;
bTh = abs(H)<thresh;
G = fft2(g20);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g20(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(24);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=20dB, thresh=0.05, ISNR = %.2f dB',ISNR))

```

```

thresh = 0.1;

```

```

bTh = abs(H)<thresh;
G = fft2(g20);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g20(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(25);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=20dB, thresh=0.1, ISNR = %.2f
dB',ISNR))

```

%Imagem SNR=30db

```

thresh = 0.000001;
bTh = abs(H)<thresh;
G = fft2(g30);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g30(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(26);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=30dB, thresh=0.000001, ISNR =
%.2f dB',ISNR))

```

```

thresh = 0.05;
bTh = abs(H)<thresh;
G = fft2(g30);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g30(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(27);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=30dB, thresh=0.05, ISNR = %.2f
dB',ISNR))

```

```

thresh = 0.1;
bTh = abs(H)<thresh;
G = fft2(g30);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g30(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(28);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=30dB, thresh=0.1, ISNR = %.2f
dB',ISNR))

```

%Imagem SNR=40db

```

thresh = 0.000001;
bTh = abs(H)<thresh;
G = fft2(g40);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g40(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(29);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=40dB, thresh=0.000001,ISNR =
%.2f dB',ISNR))

```

```

thresh = 0.05;
bTh = abs(H)<thresh;
G = fft2(g40);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g40(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(30);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=40dB, thresh=0.05,ISNR = %.2f
dB',ISNR))

```

```

thresh = 0.1;
bTh = abs(H)<thresh;
G = fft2(g40);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g40(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(31);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=40dB, thresh=0.1,ISNR = %.2f
dB',ISNR))

```

%Imagem SNR=50db

```

thresh = 0.000001;
bTh = abs(H)<thresh;
G = fft2(g50);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g50(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(32);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=50dB, thresh=0.000001,ISNR =
%.2f dB',ISNR))

```

```

thresh = 0.05;
bTh = abs(H)<thresh;
G = fft2(g50);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g50(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(33);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=50dB, thresh=0.05,ISNR = %.2f
dB',ISNR))

```

```

thresh = 0.1;
bTh = abs(H)<thresh;
G = fft2(g50);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g50(:)].^2)/sum([f(:)-fr(:)].^2) );

```

```
figure(34);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=50dB, thresh=0.1,ISNR = %.2f dB',ISNR))
```

```
%Imagem SNR=60db
```

```
thresh = 0.000001;
bTh = abs(H)<thresh;
G = fft2(g60);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g60(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(35);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=60dB, thresh=0.000001, ISNR = %.2f dB',ISNR))
```

```
thresh = 0.05;
bTh = abs(H)<thresh;
G = fft2(g60);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g60(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(36);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=60dB, thresh=0.05, ISNR = %.2f dB',ISNR))
```

```
thresh = 0.1;
bTh = abs(H)<thresh;
G = fft2(g60);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g60(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(37);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=60dB, thresh=0.1, ISNR = %.2f dB',ISNR))
```

```
%Imagem SNR=70db
```

```
thresh = 0.0000001;
bTh = abs(H)<thresh;
G = fft2(g70);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g70(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(38);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=70dB,thresh=0.000001, ISNR = %.2f dB',ISNR))
```

```
thresh = 0.05;
bTh = abs(H)<thresh;
G = fft2(g70);
FR = G./H;
FR(bTh) = 0;
```

```

fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g70(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(39);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=70dB,thresh=0.05, ISNR = %.2f
dB',ISNR))

```

```

thresh = 0.1;
bTh = abs(H)<thresh;
G = fft2(g70);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g70(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(40);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=70dB,thresh=0.1, ISNR = %.2f
dB',ISNR))

```

%Imagem SNR=80db

```

thresh = 0.000001;
bTh = abs(H)<thresh;
G = fft2(g80);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g80(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(41);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=80dB, thresh=0.000001, ISNR =
%.2f dB',ISNR))

```

```

thresh = 0.05;
bTh = abs(H)<thresh;
G = fft2(g80);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g80(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(42);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=80dB, thresh=0.000001, ISNR =
%.2f dB',ISNR))

```

```

thresh = 0.1;
bTh = abs(H)<thresh;
G = fft2(g80);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g80(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(43);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=80dB, thresh=0.000001, ISNR =
%.2f dB',ISNR))

```

%Imagem SNR=90db

```

thresh = 0.000001;
bTh = abs(H)<thresh;
G = fft2(g90);

```

```

FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g90(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(44);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=90dB, thresh=0.000001, ISNR = %.2f dB',ISNR))

```

```

thresh = 0.05;
bTh = abs(H)<thresh;
G = fft2(g90);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g90(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(45);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=90dB, thresh=0.05, ISNR = %.2f dB',ISNR))

```

```

thresh = 0.1;
bTh = abs(H)<thresh;
G = fft2(g90);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g90(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(46);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=90dB, thresh=0.1, ISNR = %.2f dB',ISNR))

```

%Imagem SNR=100db

```

thresh = 0.000001;
bTh = abs(H)<thresh;
G = fft2(g100);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g100(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(47);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=100dB, thresh=0.000001,ISNR = %.2f dB',ISNR))

```

```

thresh = 0.05;
bTh = abs(H)<thresh;
G = fft2(g100);
FR = G./H;
FR(bTh) = 0;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g100(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(48);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=100dB, thresh=0.05,ISNR = %.2f dB',ISNR))

```

```

thresh = 0.1;
bTh = abs(H)<thresh; ,

```

```

bTh = abs(H)<thresh;
G = fft2(g100);
FR = G./H;
FR(bTh) = 0;
fr = real(ifft2(FR));
ISNR = 10*log10( sum([f(:)-g100(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(48);imshow(fr,[]);title(sprintf('Rest Inv Img SNR=100dB, thresh=0.1,ISNR = %.2f
dB',ISNR))

```

%FILTRO WIENER

```

B = fft2(hc);

```

%Imagem Original

```

Sn = 0;
Sf = abs(fft2(f)).^2 ./ (256^2);
H = conj(B)./( abs(B).^2.+Sn./Sf );
G = fft2(g);
FR = G.*H;
fr = real(ifft2(FR));
ISNR = 10*log10( sum([f(:)-g(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(49);imshow(fr,[]); title(sprintf('Rest Wiener Img Original, ISNR = %.2f dB',ISNR))

```

%Imagem com SNR=5dB

```

Sn = sigma5^2;
Sf = abs(fft2(f)).^2 ./ (256^2);
H = conj(B)./( abs(B).^2.+Sn./Sf );
G = fft2(g5);
FR = G.*H;
fr = real(ifft2(FR));
ISNR = 10*log10( sum([f(:)-g5(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(50);imshow(fr,[]); title(sprintf('Rest Wiener Img SNR=5dB, ISNR = %.2f dB',ISNR))

```

%Imagem com SNR=10dB

```

Sn = sigma10^2;
Sf = abs(fft2(f)).^2 ./ (256^2);
H = conj(B)./( abs(B).^2.+Sn./Sf );
G = fft2(g10);
FR = G.*H;
fr = real(ifft2(FR));
ISNR = 10*log10( sum([f(:)-g10(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(51);imshow(fr,[]); title(sprintf('Rest Wiener Img SNR=10dB, ISNR = %.2f
dB',ISNR))

```

%Imagem com SNR=20dB

```

Sn = sigma20^2;
Sf = abs(fft2(f)).^2 ./ (256^2);
H = conj(B)./( abs(B).^2.+Sn./Sf );
G = fft2(g20);
FR = G.*H;

```

```

fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g20(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(52);imshow(fr,[]); title(sprintf('Rest Wiener Img SNR=20dB, ISNR = %.2f
dB',ISNR))

```

%Imagem com SNR=30dB

```

Sn = sigma30^2;
Sf = abs(fft2(f)).^2 ./ (256^2);
H = conj(B)./( abs(B).^2.+Sn./Sf );
G = fft2(g30);
FR = G.*H;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g30(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(53);imshow(fr,[]); title(sprintf('Rest Wiener Img SNR=30dB, ISNR = %.2f
dB',ISNR))

```

%Imagem com SNR=40dB

```

Sn = sigma40^2;
Sf = abs(fft2(f)).^2 ./ (256^2);
H = conj(B)./( abs(B).^2.+Sn./Sf );
G = fft2(g40);
FR = G.*H;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g40(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(54);imshow(fr,[]); title(sprintf('Rest Wiener Img SNR=40dB, ISNR = %.2f
dB',ISNR))

```

%Imagem com SNR=50dB

```

Sn = sigma50^2;
Sf = abs(fft2(f)).^2 ./ (256^2);
H = conj(B)./( abs(B).^2.+Sn./Sf );
G = fft2(g50);
FR = G.*H;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g50(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(55);imshow(fr,[]); title(sprintf('Rest Wiener Img SNR=50dB, ISNR = %.2f
dB',ISNR))

```

%Imagem com SNR=60dB

```

Sn = sigma60^2;
Sf = abs(fft2(f)).^2 ./ (256^2);
H = conj(B)./( abs(B).^2.+Sn./Sf );
G = fft2(g60);
FR = G.*H;
fr = real(iff2(FR));
ISNR = 10*log10( sum([f(:)-g60(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(56);imshow(fr,[]); title(sprintf('Rest Wiener Img SNR=60dB, ISNR = %.2f
dB',ISNR))

```

%Imagem com SNR=70dB


```

Sn = sigma70^2;
Sf = abs(fft2(f)).^2 ./ (256^2);
H = conj(B)./( abs(B).^2.+Sn./Sf );
G = fft2(g70);
FR = G.*H;
fr = real(ifft2(FR));
ISNR = 10*log10( sum([f(:)-g70(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(57);imshow(fr,[]); title(sprintf('Rest Wiener Img SNR=70dB, ISNR = %.2f
dB',ISNR))

```

%Imagem com SNR=80dB

```

Sn = sigma80^2;
Sf = abs(fft2(f)).^2 ./ (256^2);
H = conj(B)./( abs(B).^2.+Sn./Sf );
G = fft2(g80);
FR = G.*H;
fr = real(ifft2(FR));
ISNR = 10*log10( sum([f(:)-g80(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(58);imshow(fr,[]); title(sprintf('Rest Wiener Img SNR=80dB, ISNR = %.2f
dB',ISNR))

```

%Imagem com SNR=90dB

```

Sn = sigma90^2;
Sf = abs(fft2(f)).^2 ./ (256^2);
H = conj(B)./( abs(B).^2.+Sn./Sf );
G = fft2(g90);
FR = G.*H;
fr = real(ifft2(FR));
ISNR = 10*log10( sum([f(:)-g90(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(59);imshow(fr,[]); title(sprintf('Rest Wiener Img SNR=90dB, ISNR = %.2f
dB',ISNR))

```

%Imagem com SNR=100dB

```

Sn = sigma100^2;
Sf = abs(fft2(f)).^2 ./ (256^2);
H = conj(B)./( abs(B).^2.+Sn./Sf );
G = fft2(g100);
FR = G.*H;
fr = real(ifft2(FR));
ISNR = 10*log10( sum([f(:)-g100(:)].^2)/sum([f(:)-fr(:)].^2) );
figure(60);imshow(fr,[]); title(sprintf('Rest Wiener Img SNR=100dB, ISNR = %.2f
dB',ISNR))

```

ANEXO B – Imagem original e degradadas

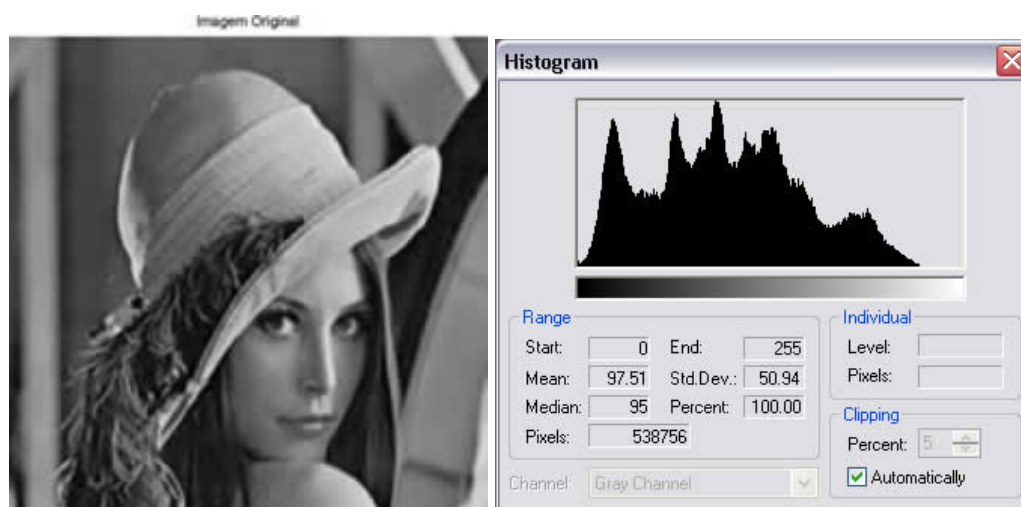


Figura B.1 – Imagem original e seu histograma.



Figura B.2 – Imagem original embaçada e seu histograma.

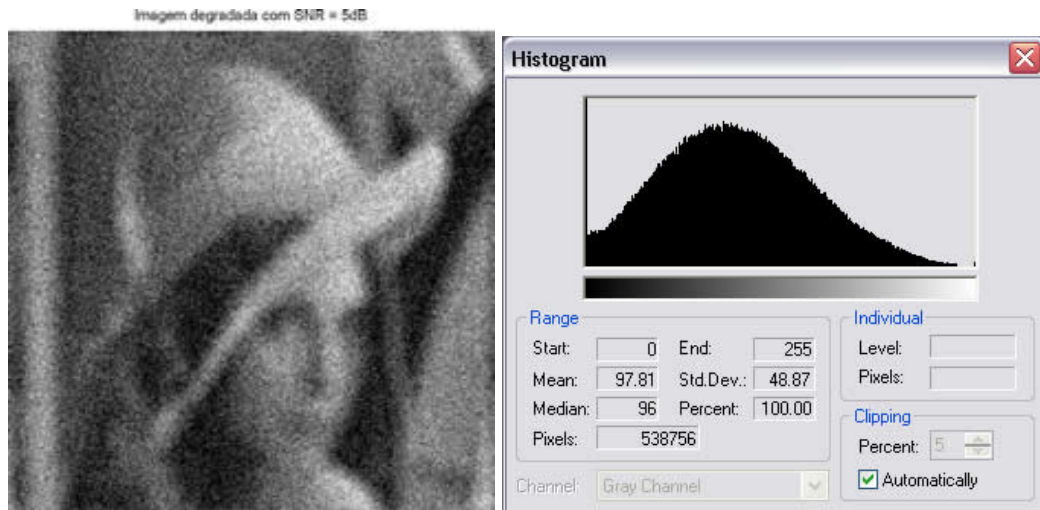


Figura B.3 – Imagem degradada com SNR = 5 dB e seu histograma.

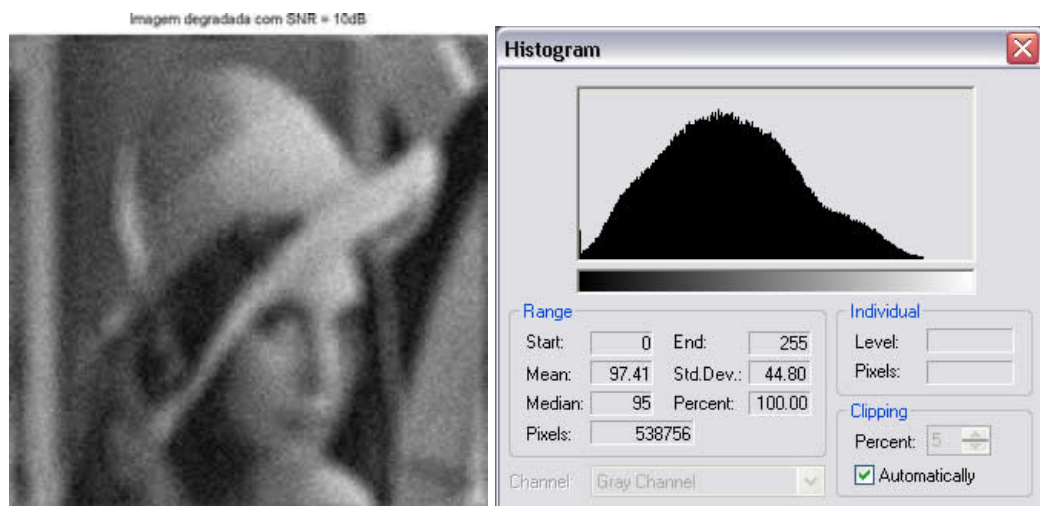


Figura B.4 – Imagem degradada com SNR = 10 dB e seu histograma.



Figura B.5 – Imagem degradada com SNR = 20 dB e seu histograma.

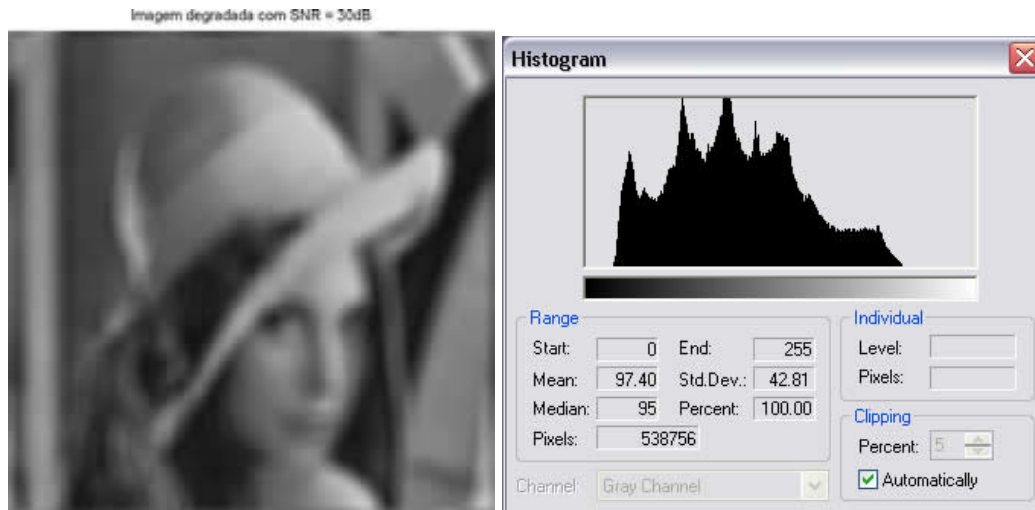


Figura B.6 – Imagem degradada com SNR = 30 dB e seu histograma.

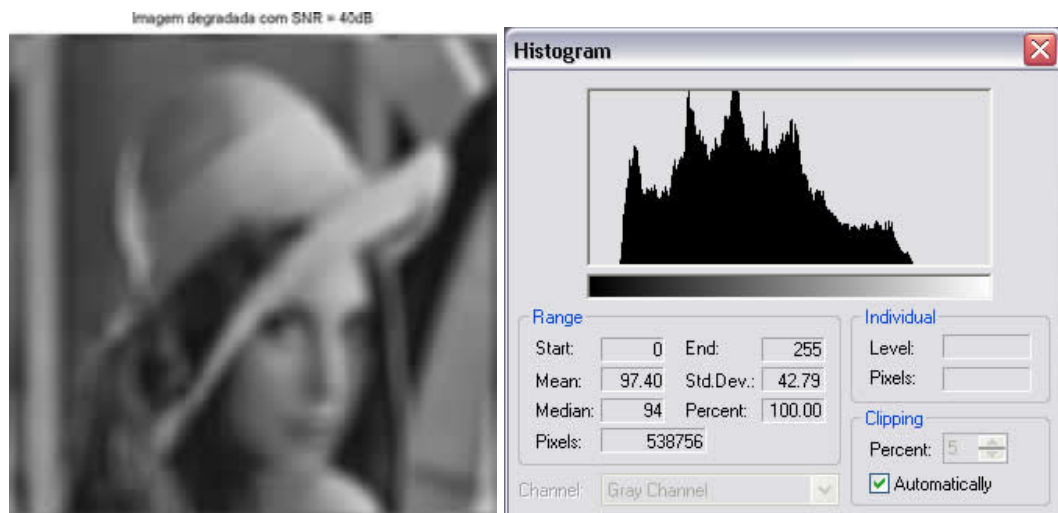


Figura B.7 – Imagem degradada com SNR = 40 dB e seu histograma.

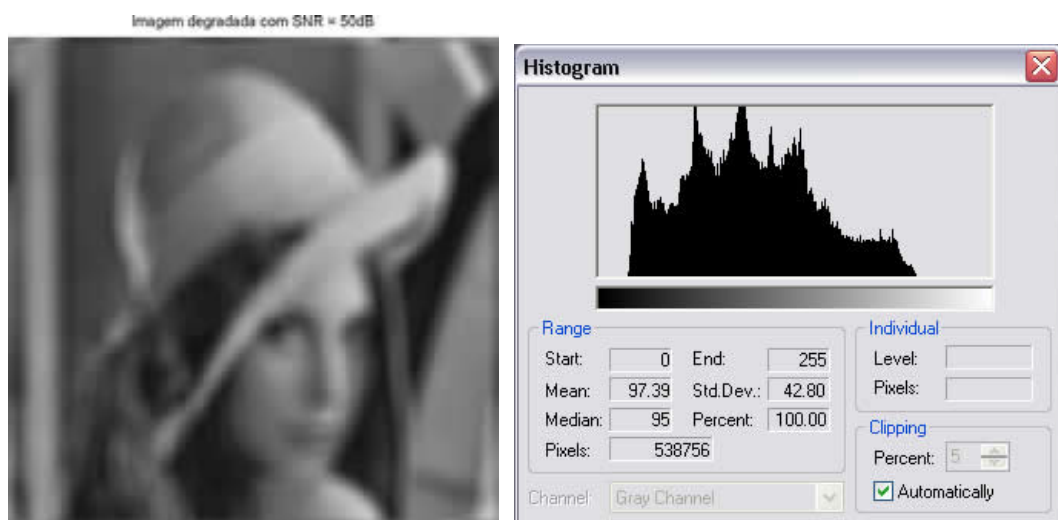


Figura B.8 – Imagem degradada com SNR = 50 dB e seu histograma.



Figura B.9 – Imagem degradada com SNR = 60 dB e seu histograma.

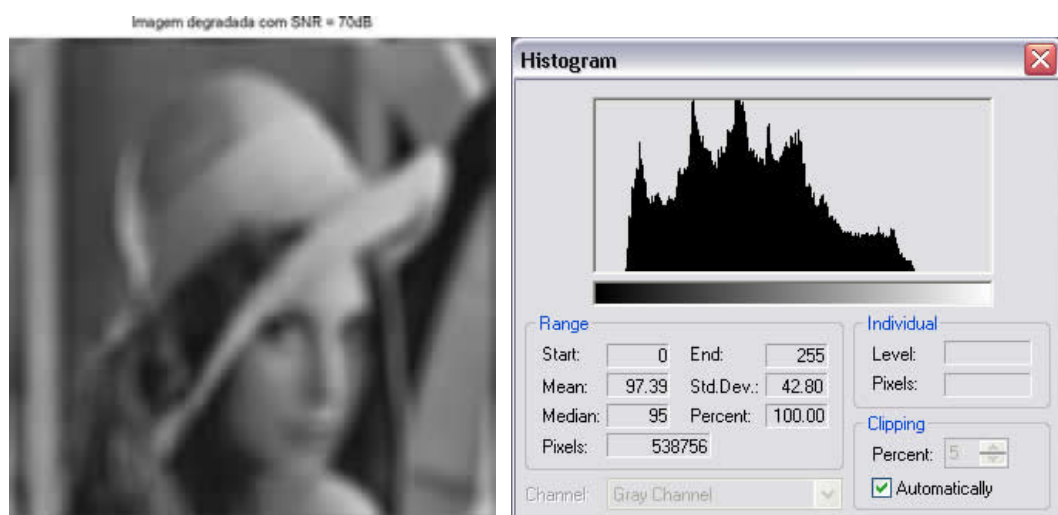


Figura B.10 – Imagem degradada com SNR = 70 dB e seu histograma.

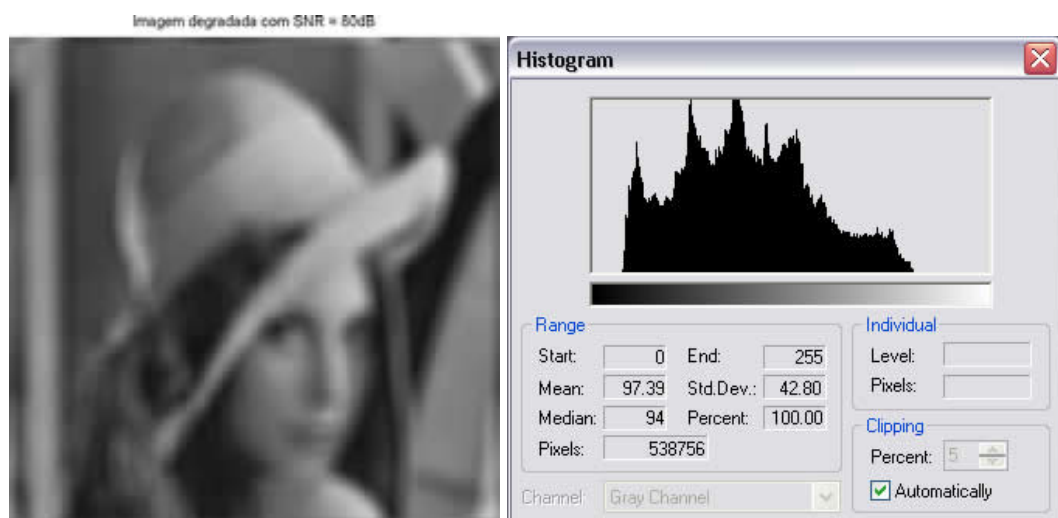


Figura B.11 – Imagem degradada com SNR = 80 dB e seu histograma.

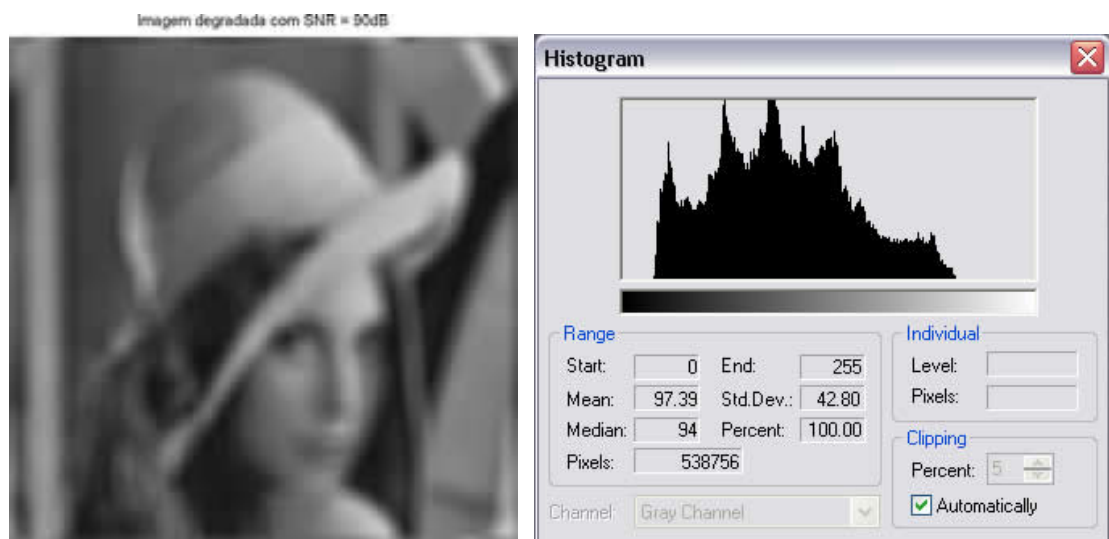


Figura B.12 – Imagem degradada com SNR = 90 dB e seu histograma.

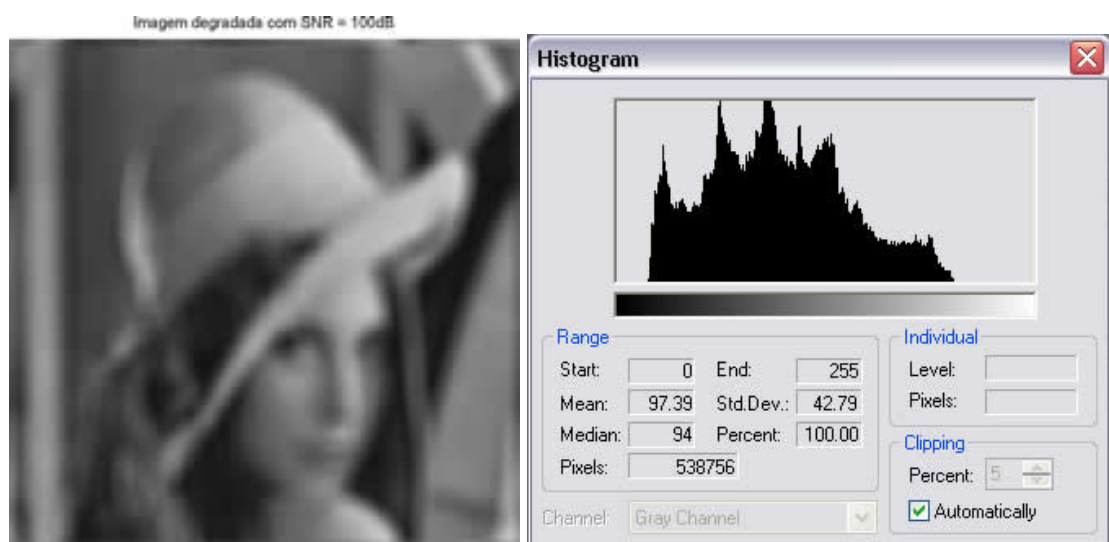


Figura B.13 – Imagem degradada com SNR = 100 dB e seu histograma.

ANEXO C – Filtragem inversa

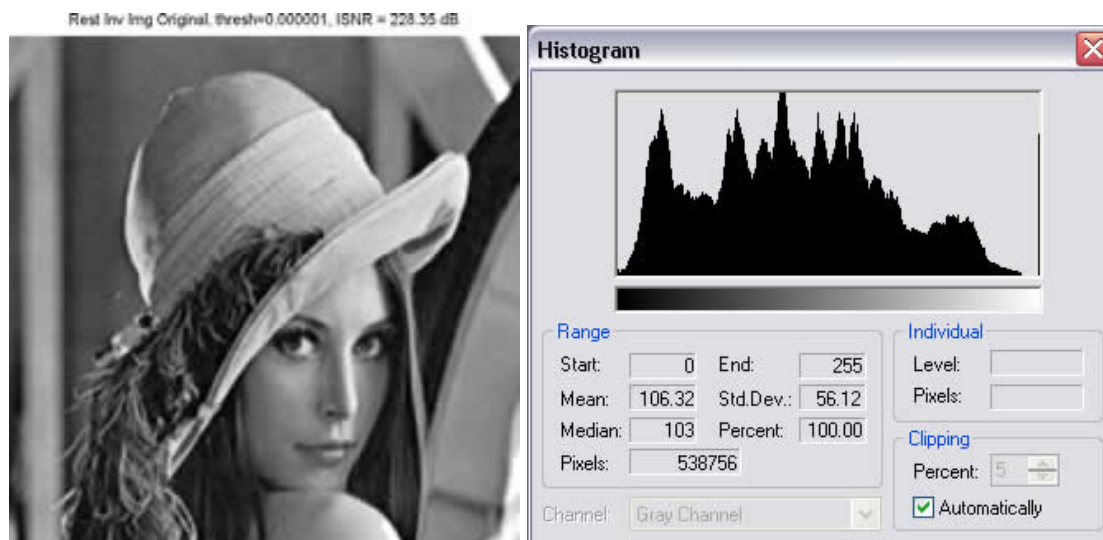


Figura C.1 – Filt. inversa, Imagem Original, SNR=0dB, thresh=0.000001, ISNR= 228.35dB e seu histograma.

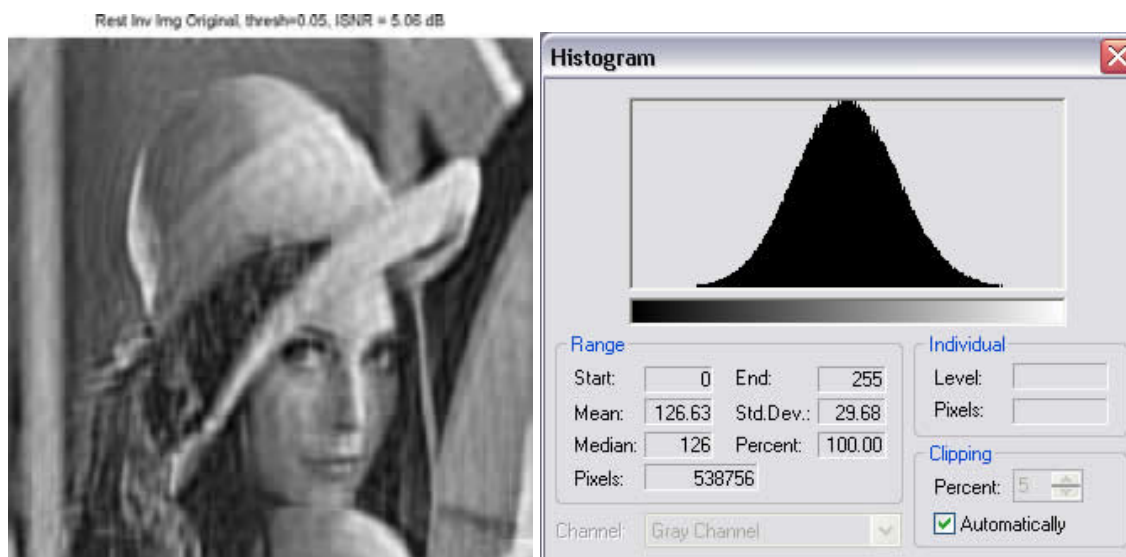


Figura C.2 – Filt. inversa, Imagem Original, SNR=0dB, thresh=0.05, ISNR= 5.06dB e seu histograma.

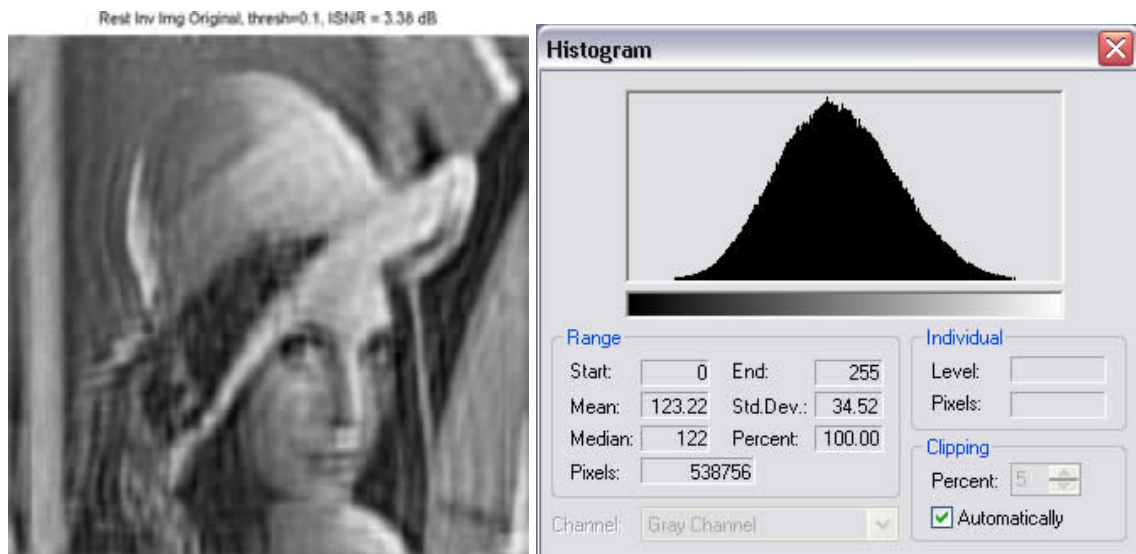


Figura C.3 – Filt. inversa, Imagem Original, SNR=0dB, thresh=0.1, ISNR= 3.38dB e seu histograma.

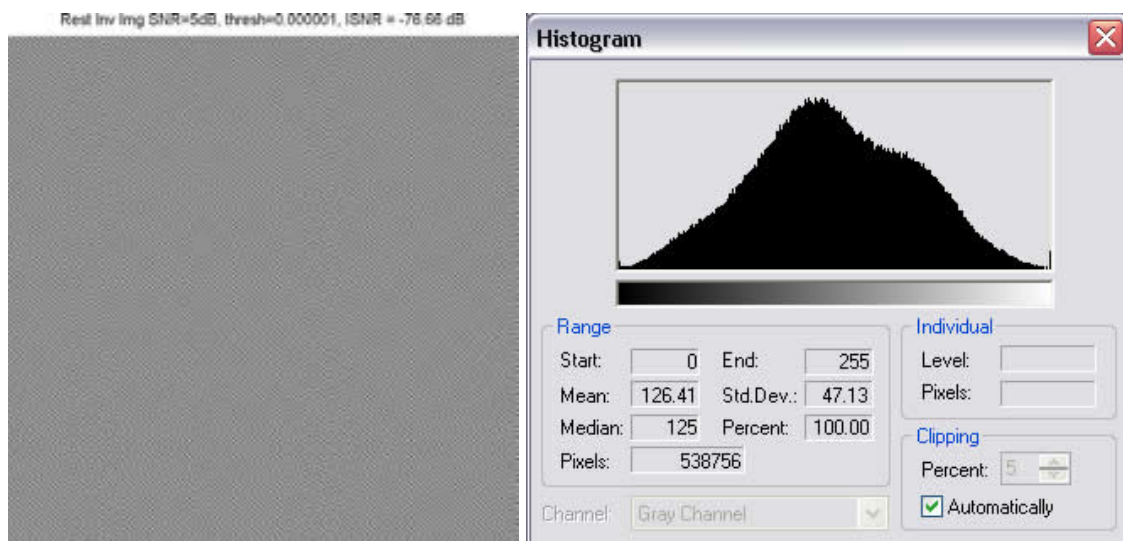


Figura C.4 – Filt. inversa, Embaçam.+ SNR=5dB, thresh=0.000001, ISNR= -76.66dB e seu histograma.

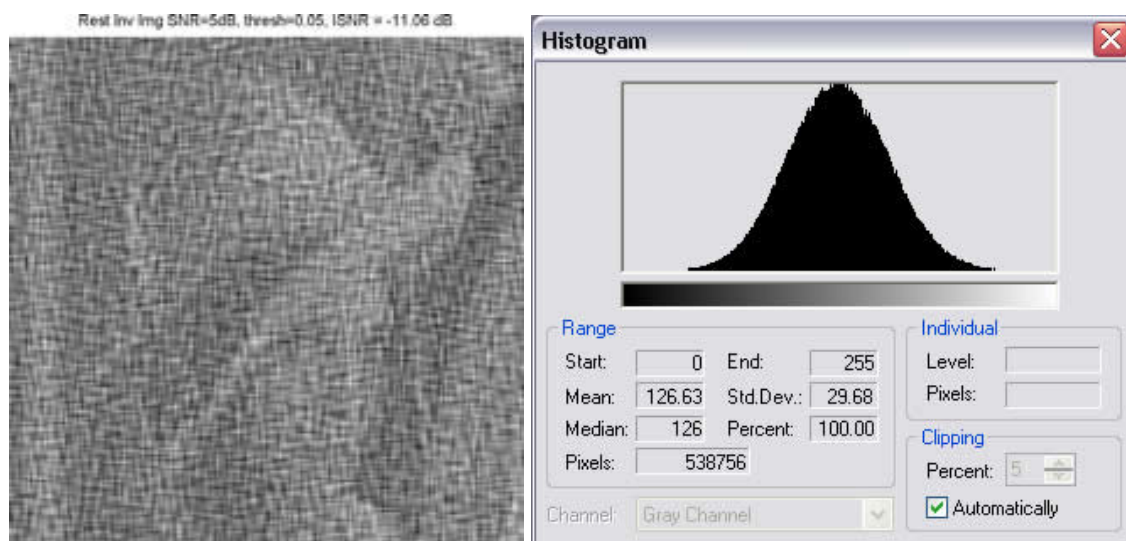


Figura C.5 – Filt. inversa, Embaçam.+ SNR=5dB, thresh=0.05, ISNR= -11.06dB e seu histograma.

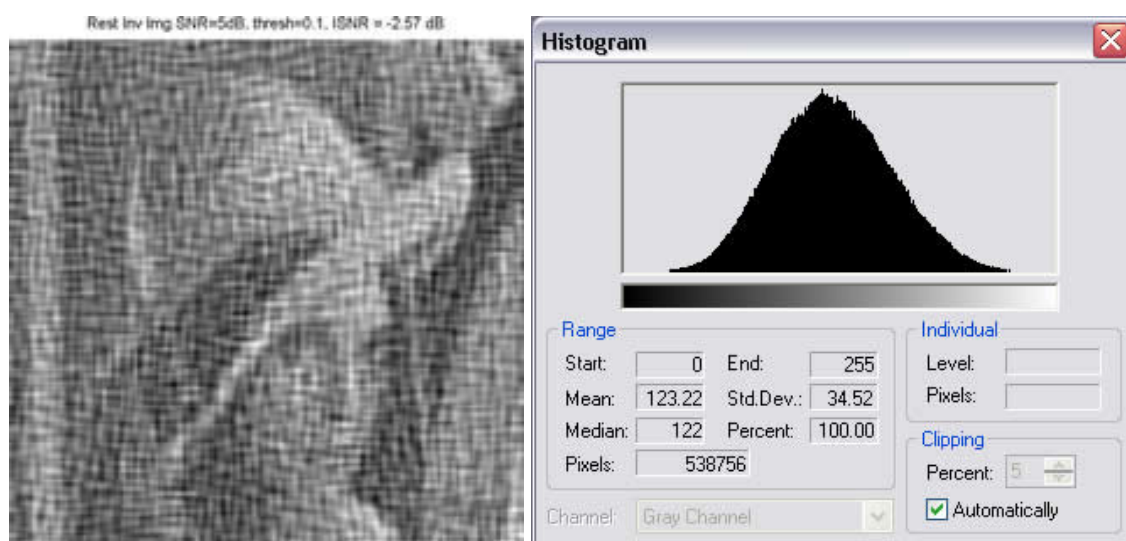


Figura C.6 – Filt. inversa, Embaçam.+ SNR=5dB, thresh=0.1, ISNR= -2.57dB e seu histograma.

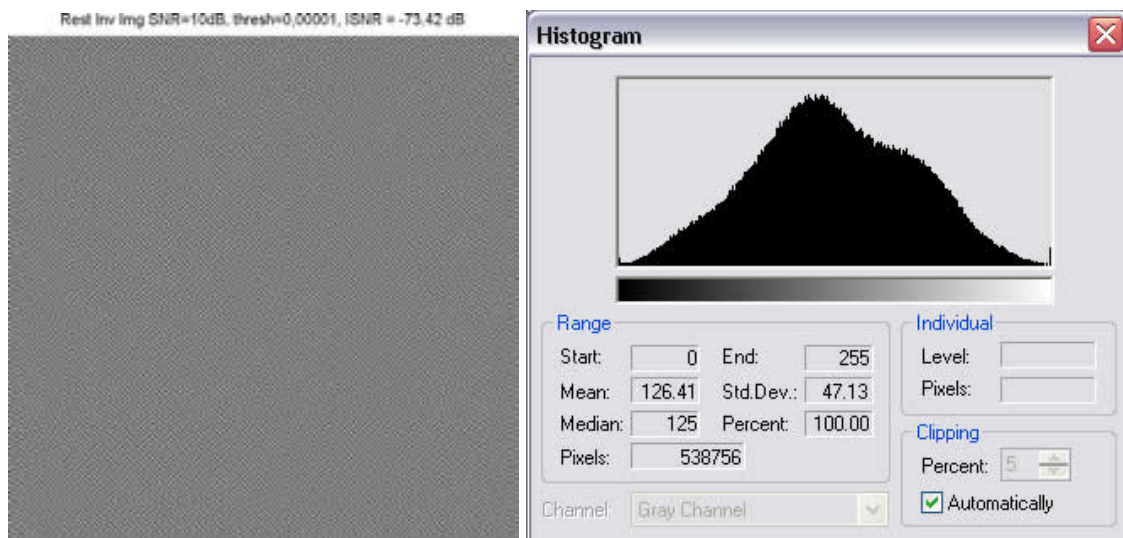


Figura C.7 – Filt. inversa, Embaçam.+ SNR=10dB, thresh=0.000001, ISNR= -73.42dB e seu histograma.

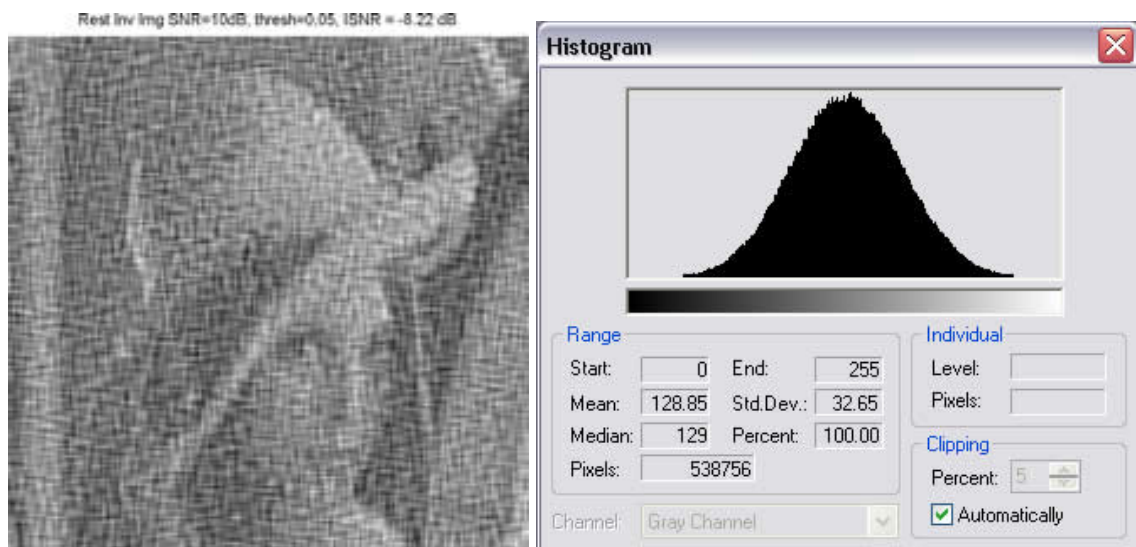


Figura C.8 – Filt. inversa, Embaçam.+ SNR=10dB, thresh=0.05, ISNR= -8.22dB e seu histograma.

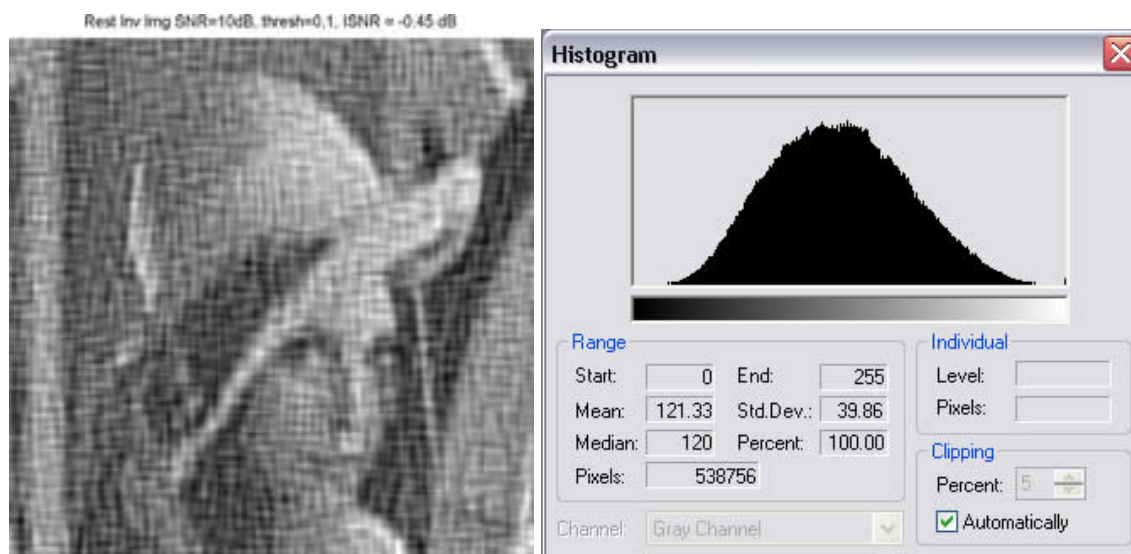


Figura C.9 – Filt. inversa, Embaçam.+ SNR=10dB, thresh=0.1, ISNR= -0.45dB e seu histograma.

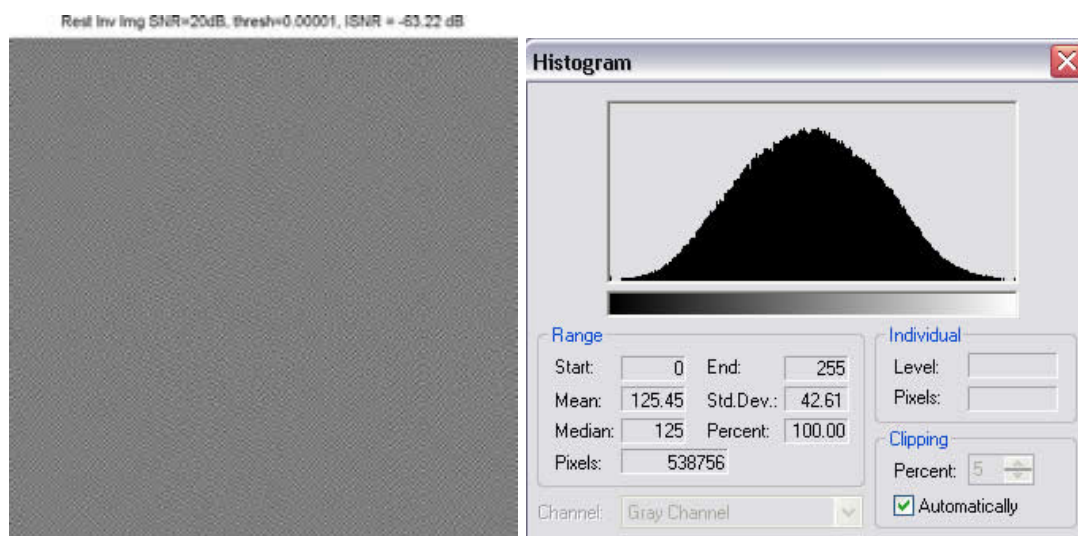


Figura C.10 – Filt. inversa, Embaçam.+ SNR=20dB, thresh=0.000001, ISNR= -63.22dB e seu histograma.

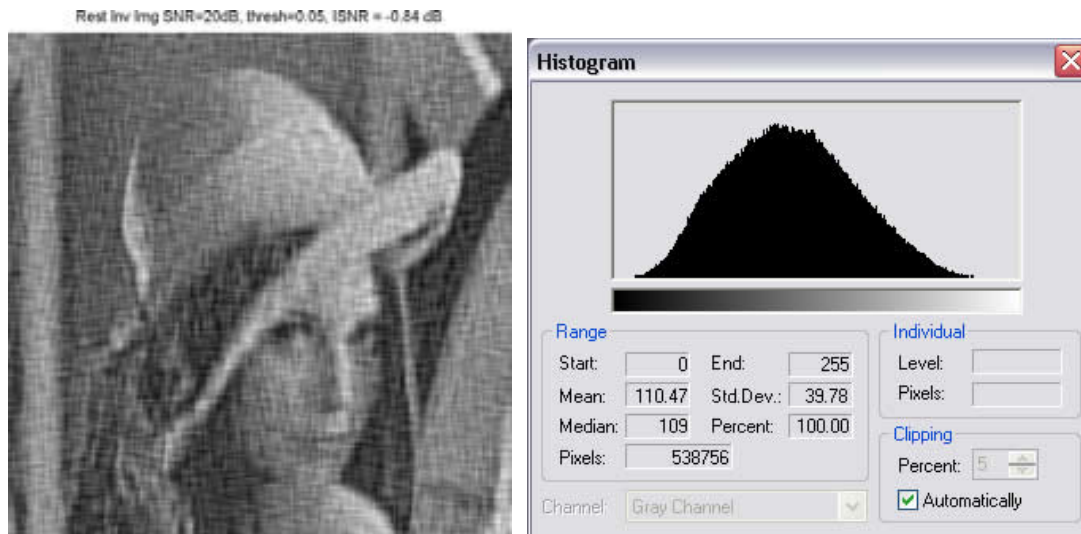


Figura C.11 – Filt. inversa, Embaçam.+ SNR=20dB, thresh=0.05, ISNR= -0.84dB e seu histograma.

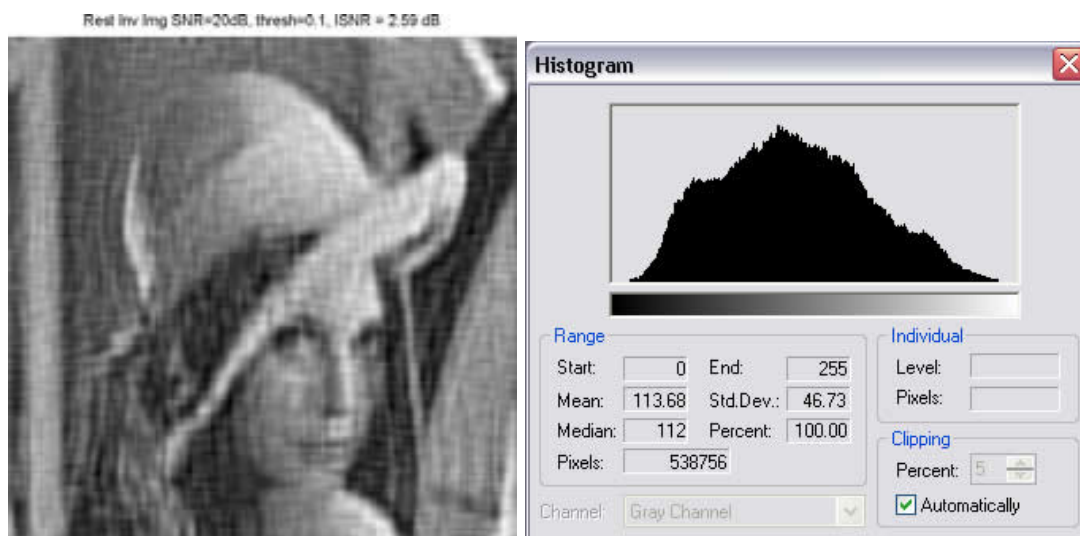


Figura C.12 – Filt. inversa, Embaçam.+ SNR=20dB, thresh=0.1, ISNR= 2.59dB e seu histograma.

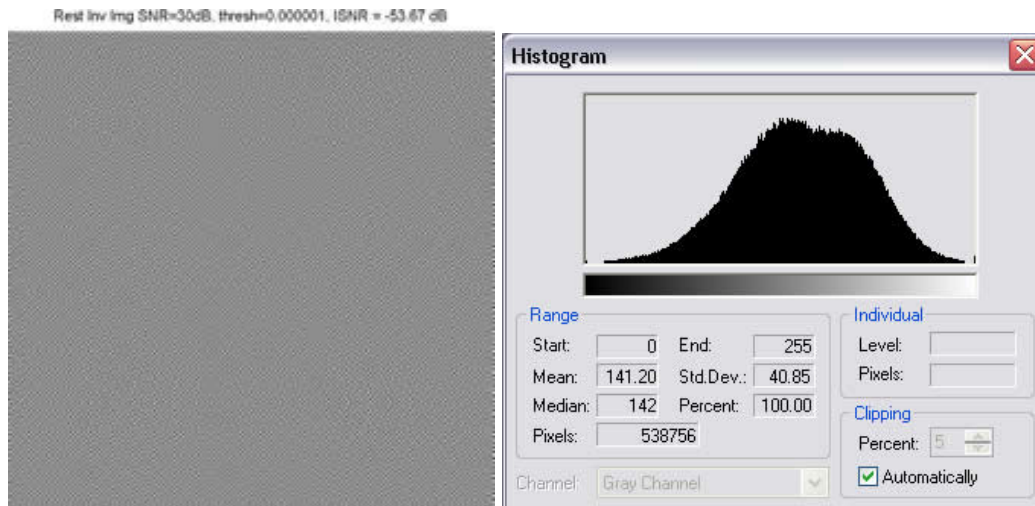


Figura C.13 – Filt. inversa, Embaçam.+ SNR=30dB, thresh=0.000001, ISNR= -53.67dB e seu histograma.

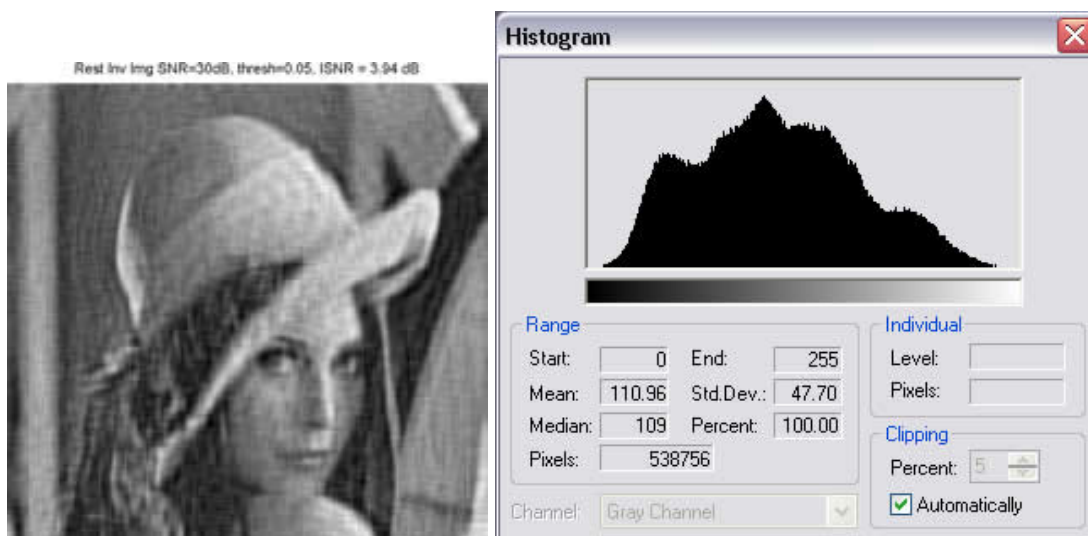


Figura C.14 – Filt. inversa, Embaçam.+ SNR=30dB, thresh=0.05, ISNR= 3.94dB e seu histograma.

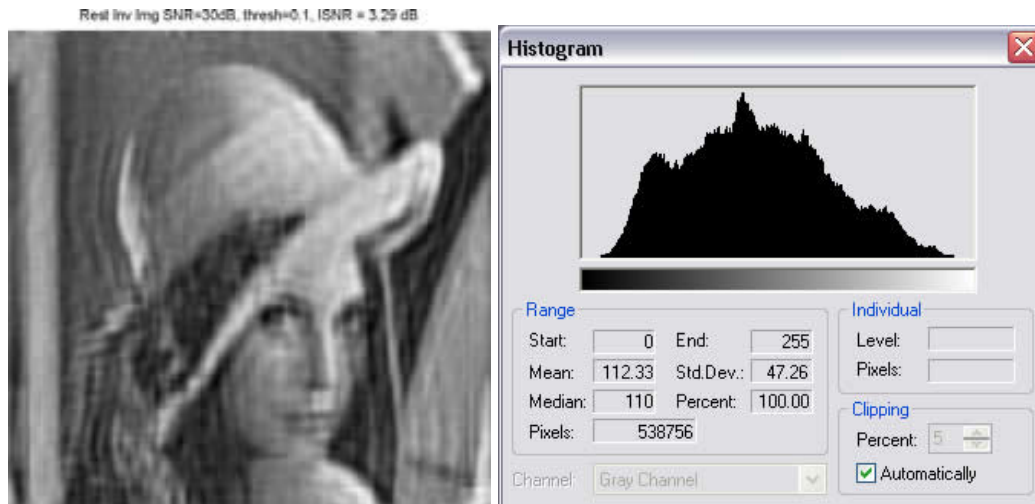


Figura C.15 – Filt. inversa, Embaçam.+ SNR=30dB, thresh=0.1, ISNR= 3.29dB e seu histograma.

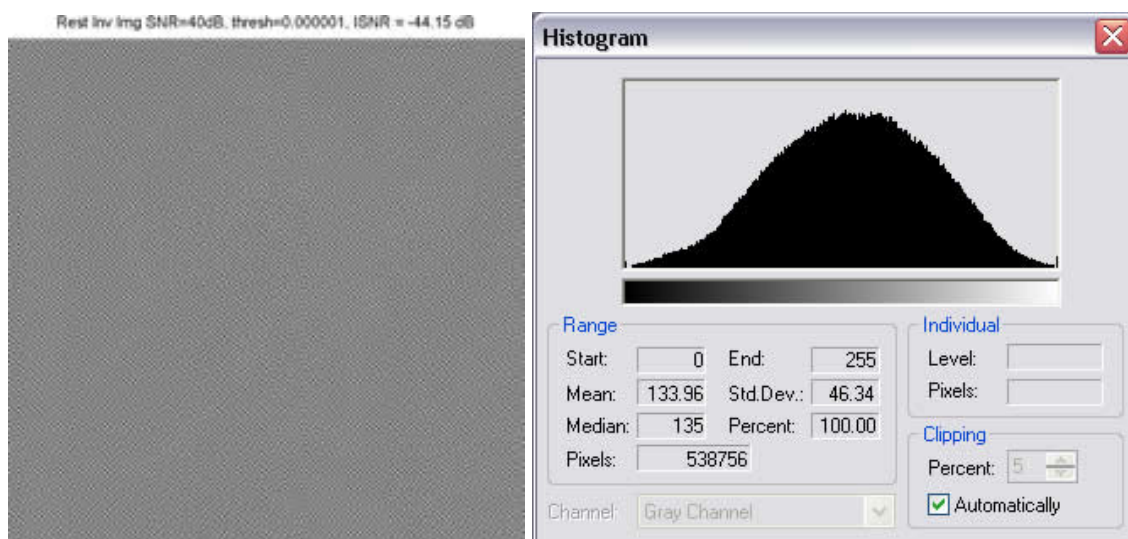


Figura C.16 – Filt. inversa, Embaçam.+ SNR=40dB, thresh=0.000001, ISNR= -44.15dB e seu histograma.

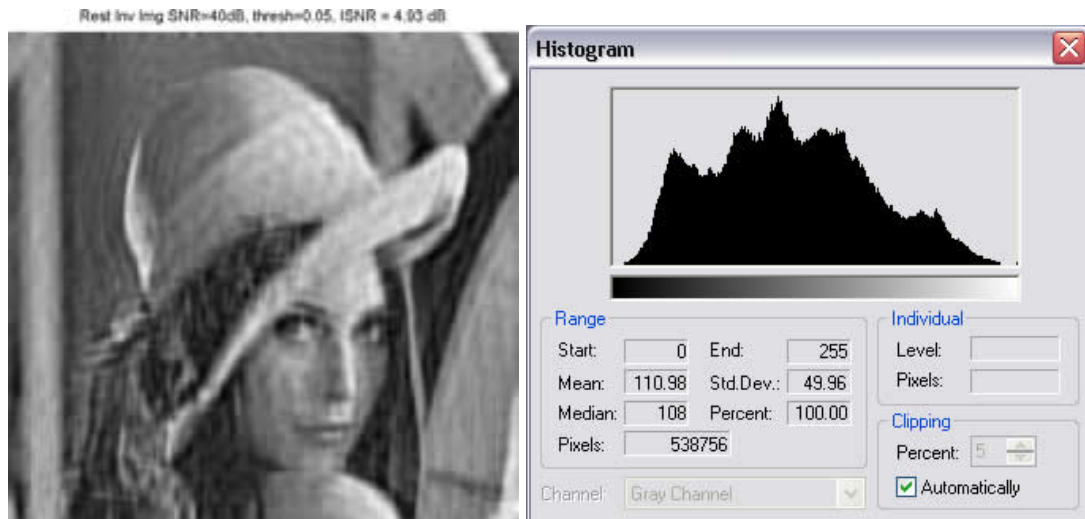


Figura C.17 – Filt. inversa, Embaçam.+ SNR=40dB, thresh=0.05, ISNR= 4.93dB e seu histograma.

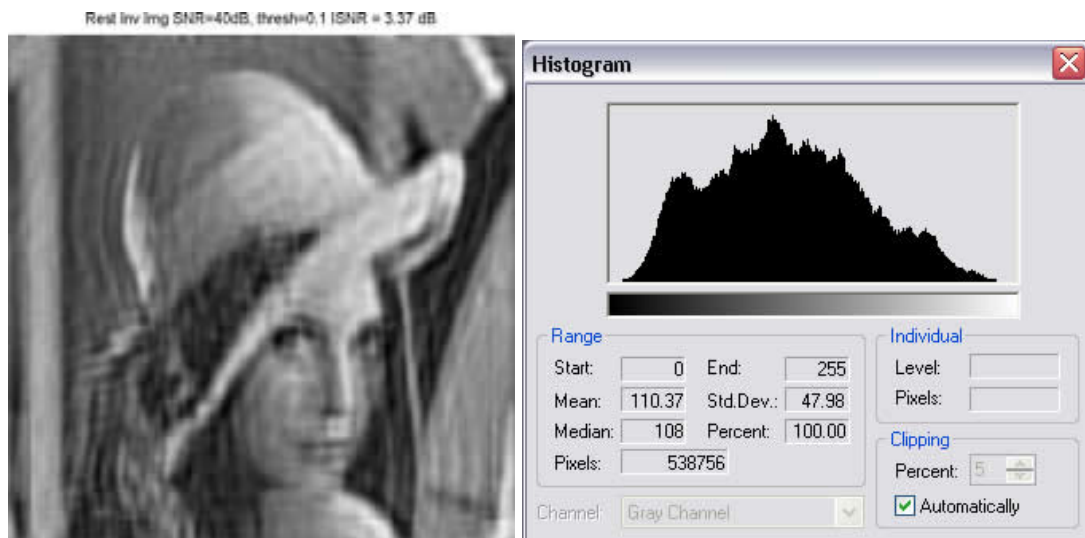


Figura C.18 – Filt. inversa, Embaçam.+ SNR=40dB, thresh=0.1, ISNR= 3.37dB e seu histograma.

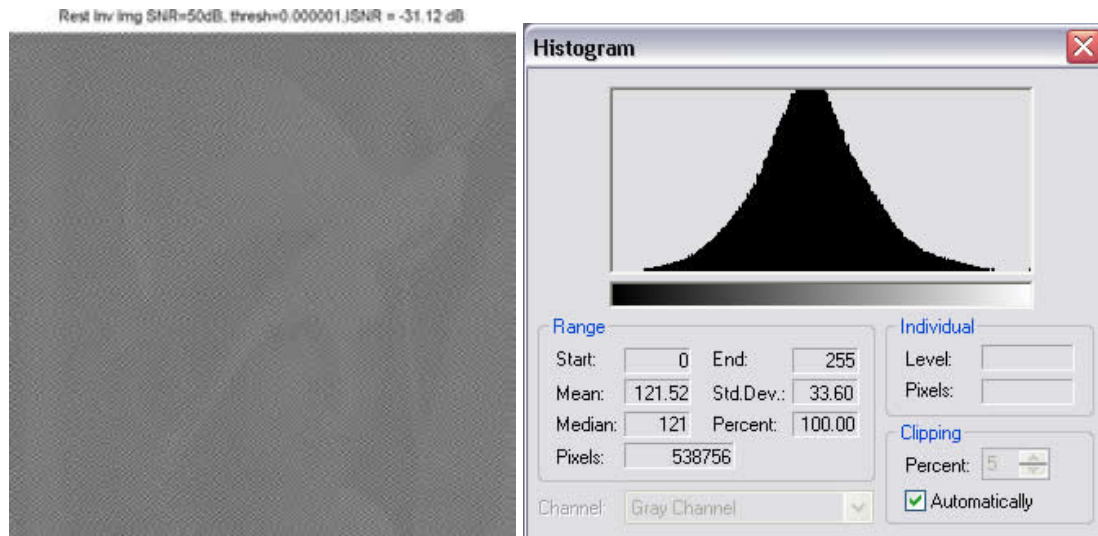


Figura C.19 – Filt. inversa, Embaçam.+ SNR=50dB, thresh=0.000001, ISNR= -31.12dB e seu histograma.

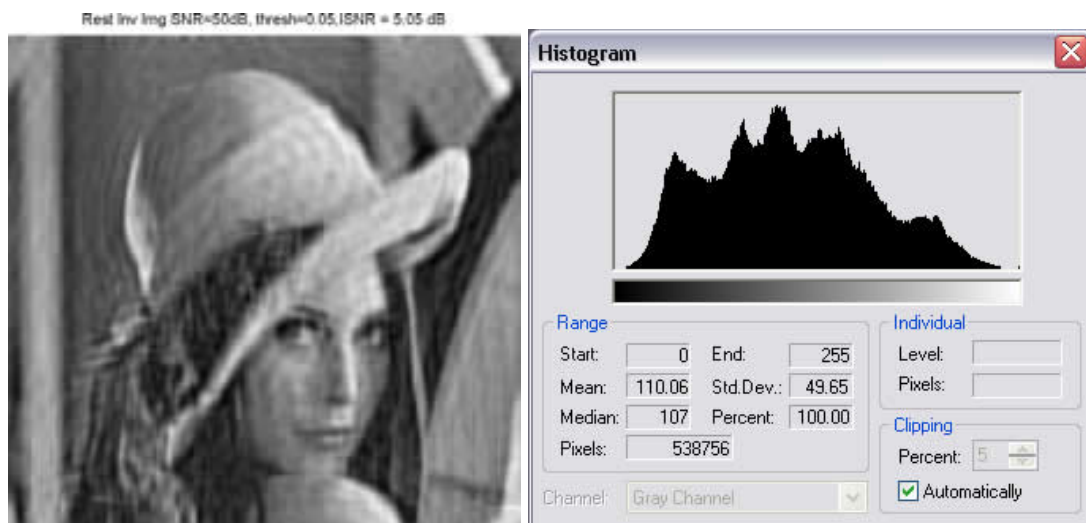


Figura C.20 – Filt. inversa, Embaçam.+ SNR=50dB, thresh=0.05, ISNR= 5.05dB e seu histograma.

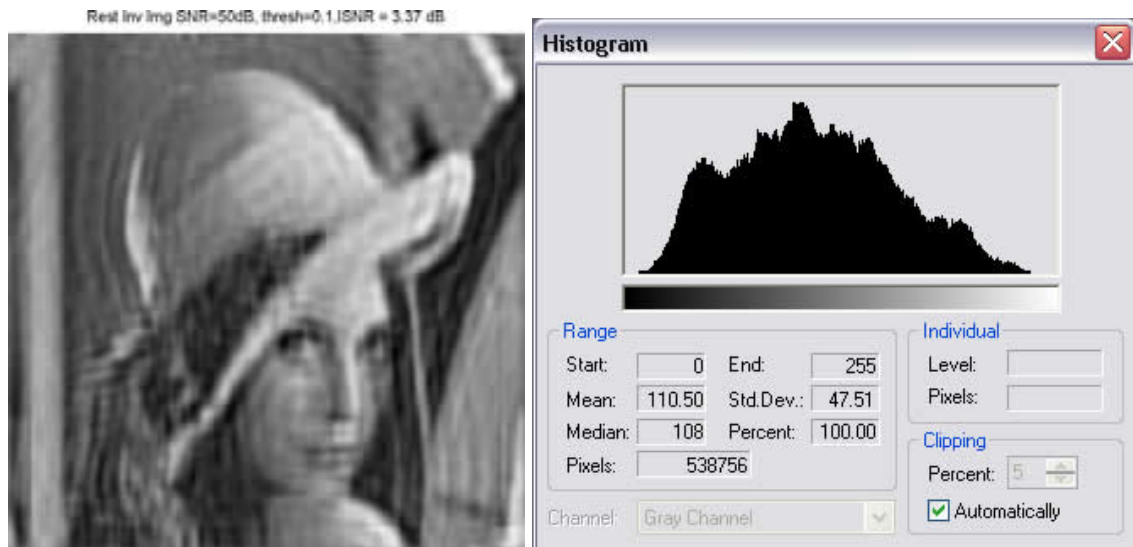


Figura C.21 – Filt. inversa, Embaçam.+ SNR=50dB, thresh=0.1, ISNR= 3.37dB e seu histograma.

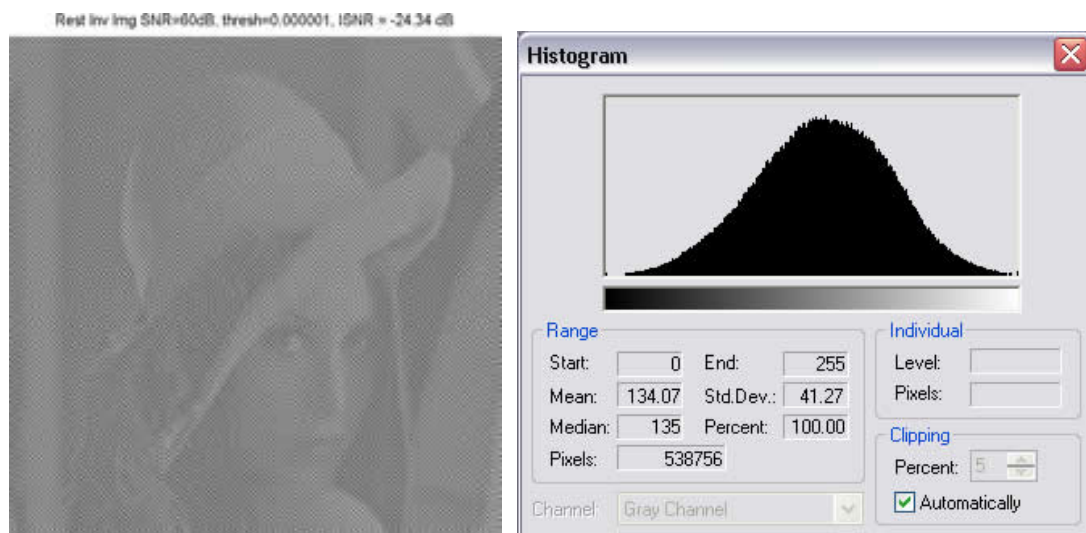


Figura C.22 – Filt. inversa, Embaçam.+ SNR=60dB, thresh=0.000001, ISNR= -24.34dB e seu histograma.

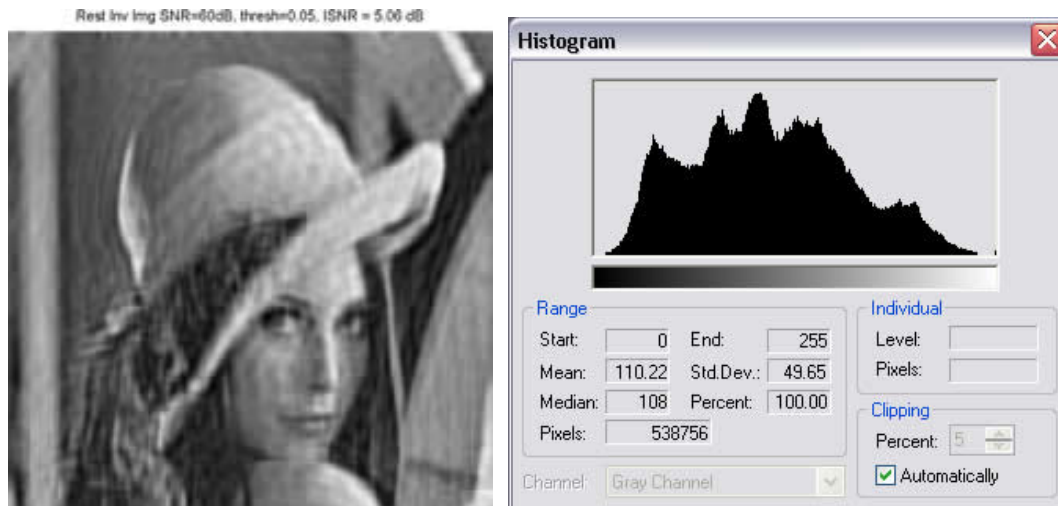


Figura C.23 – Filt. inversa, Embaçam.+ SNR=60dB, thresh=0.05, ISNR= 5.06dB e seu histograma.

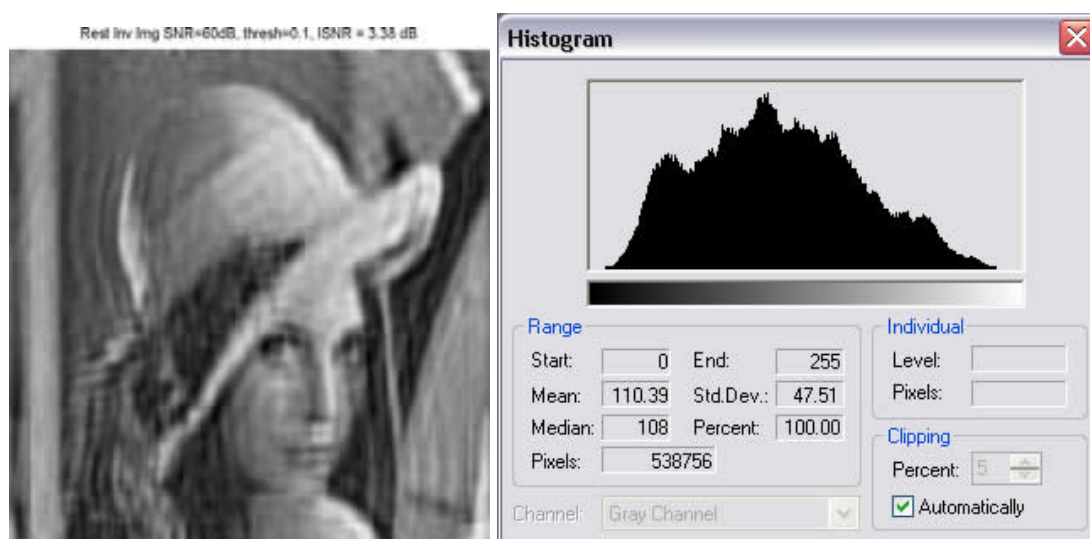


Figura C.24 – Filt. inversa, Embaçam.+ SNR=60dB, thresh=0.1, ISNR= 3.38dB e seu histograma.

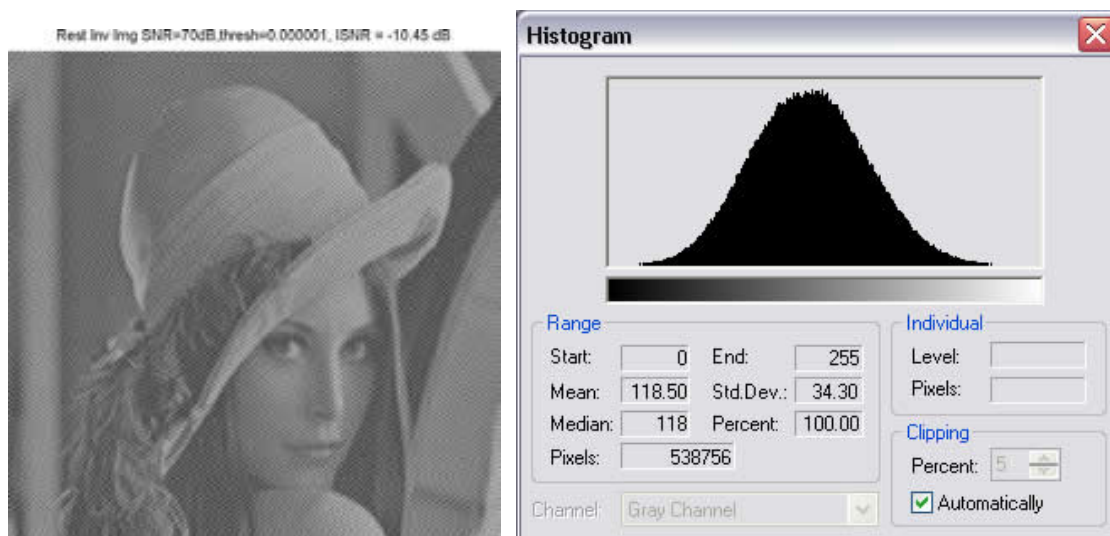


Figura C.25 – Filt. inversa, Embaçam.+ SNR=70dB, thresh=0.000001, ISNR= -10.45dB e seu histograma.

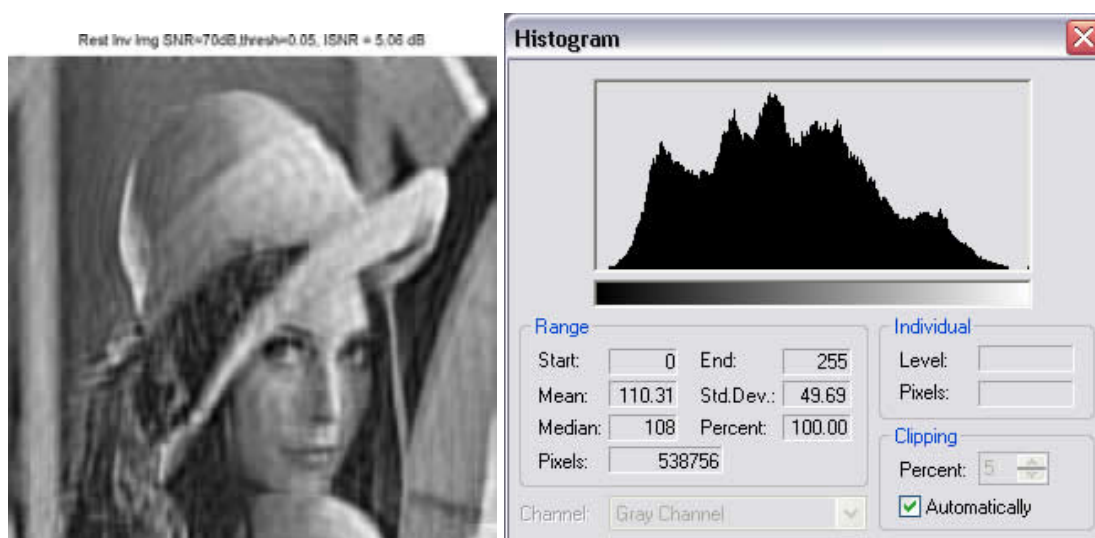


Figura C.26 – Filt. inversa, Embaçam.+ SNR=70dB, thresh=0.05, ISNR= 5.06dB e seu histograma.

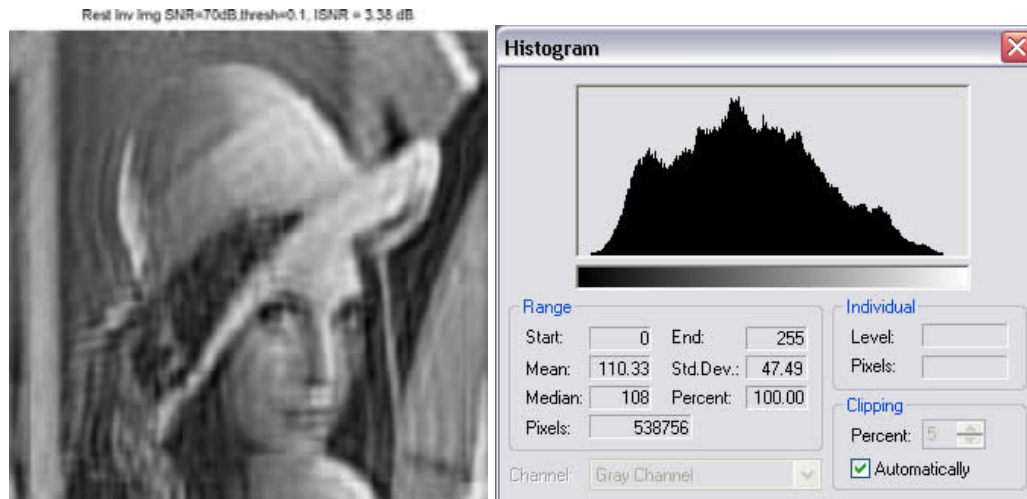


Figura C.27 – Filt. inversa, Embaçam.+ SNR=70dB, thresh=0.1, ISNR= 3.38dB e seu histograma.

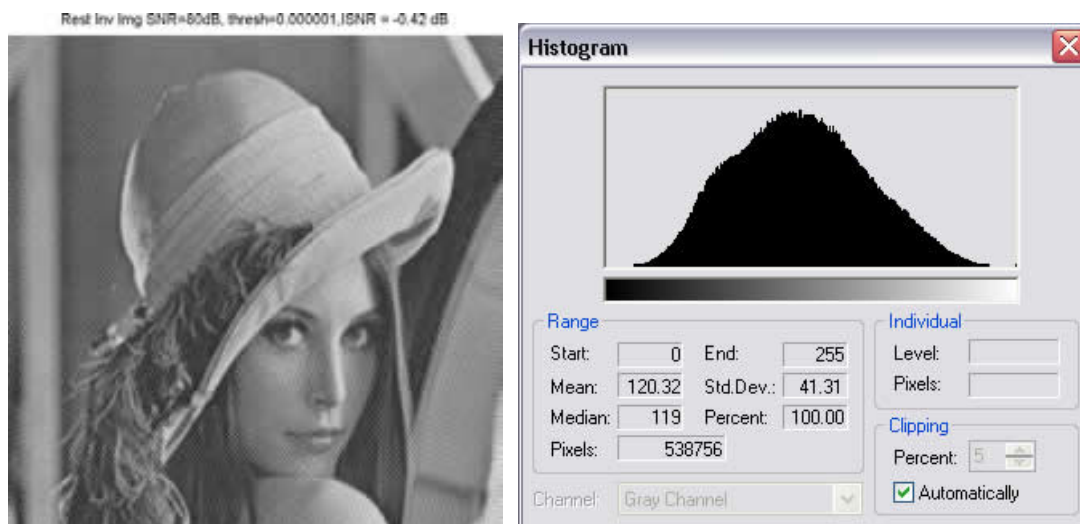


Figura C.28 – Filt. inversa, Embaçam.+ SNR=80dB, thresh=0.000001, ISNR= -0.42dB e seu histograma.

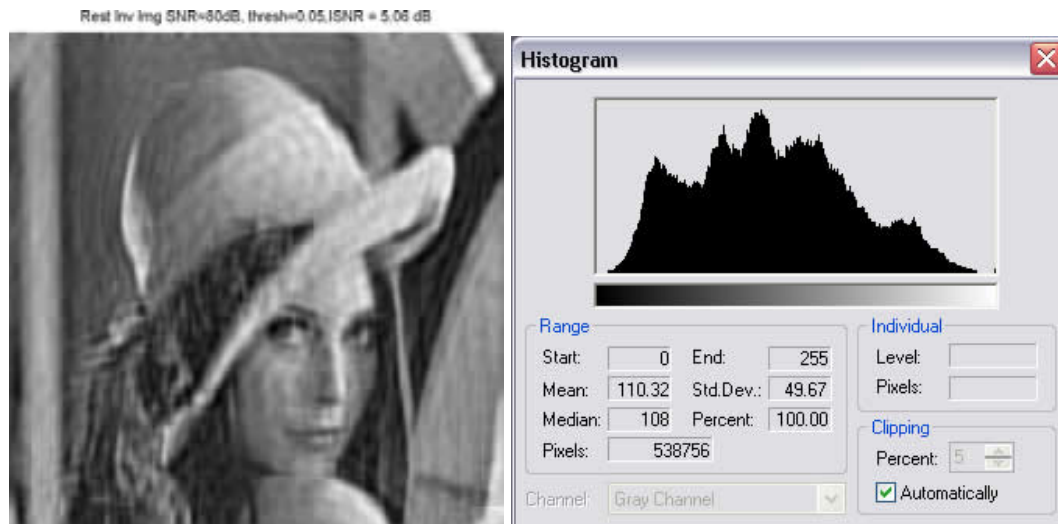


Figura C.29 – Filt. inversa, Embaçam.+ SNR=80dB, thresh=0.05, ISNR= 5.06dB e seu histograma.

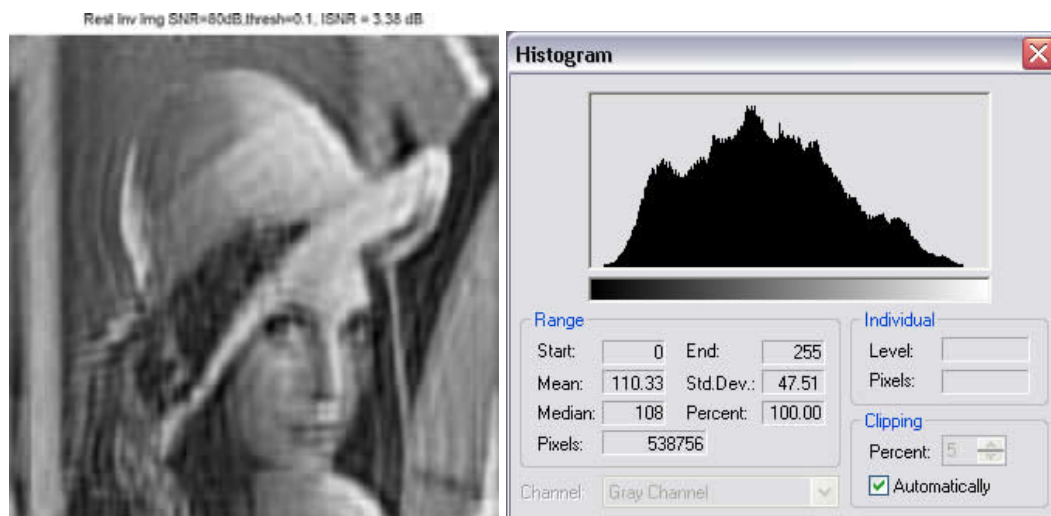


Figura C.30 – Filt. inversa, Embaçam.+ SNR=80dB, thresh=0.1, ISNR= 3.38dB e seu histograma.

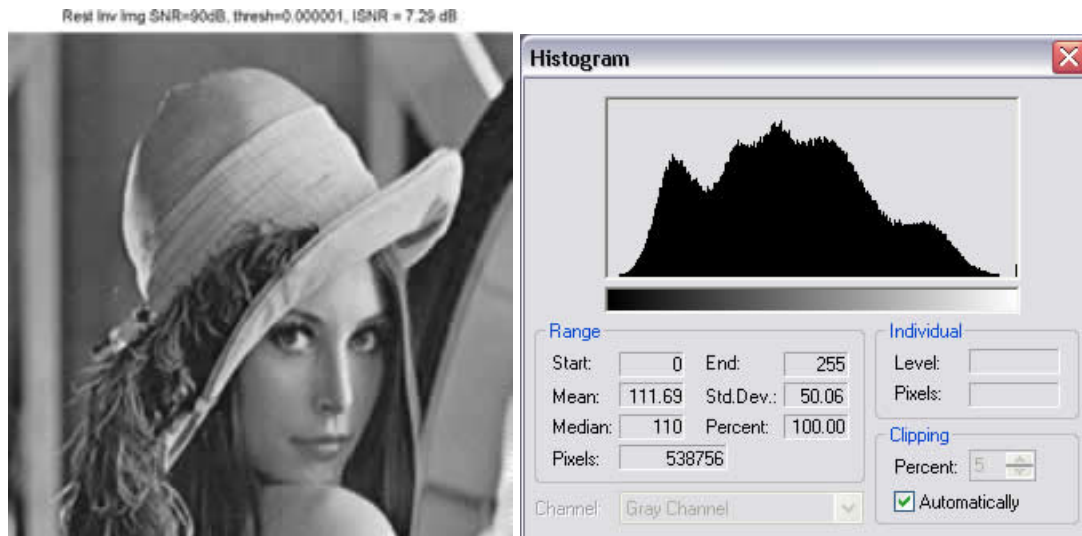


Figura C.31 – Filt. inversa, Embaçam.+ SNR=90dB, thresh=0.000001, ISNR= 7.29dB e seu histograma.

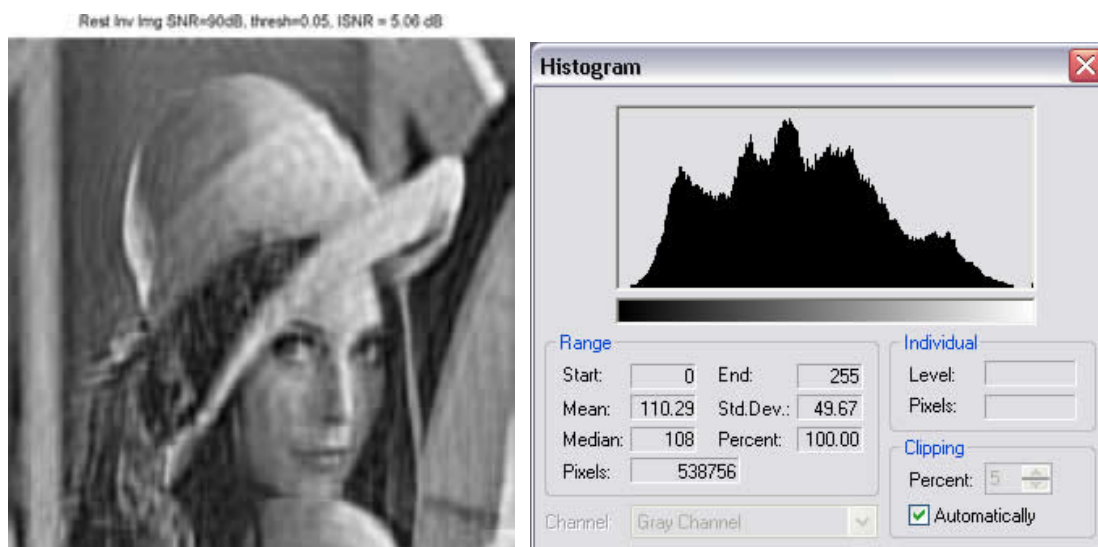


Figura C.32 – Filt. inversa, Embaçam.+ SNR=90dB, thresh=0.05, ISNR= 5.06dB e seu histograma.

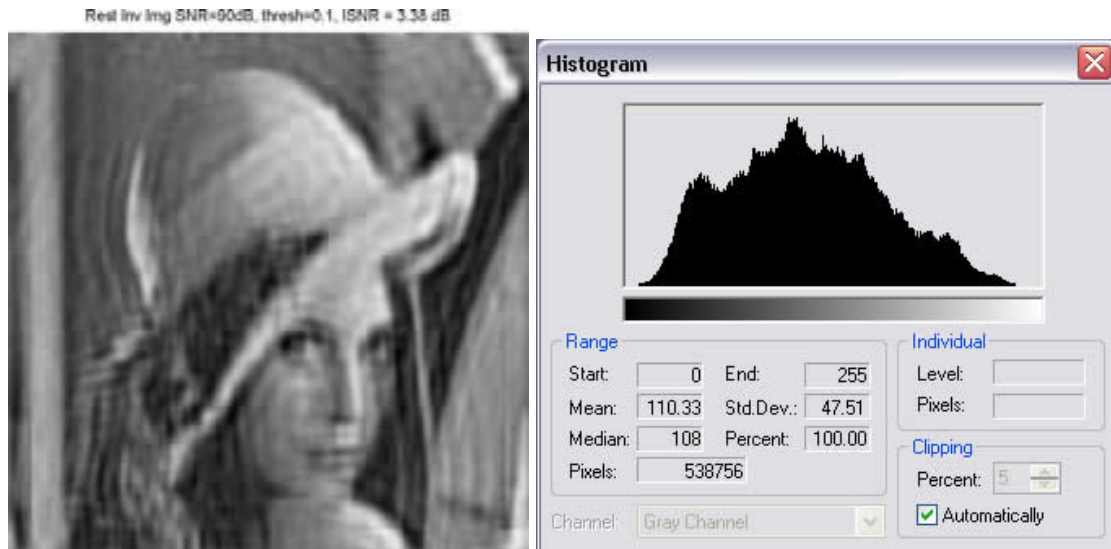


Figura C.33 – Filt. inversa, Embaçam.+ SNR=90dB, thresh=0.1, ISNR= 3.38dB e seu histograma.

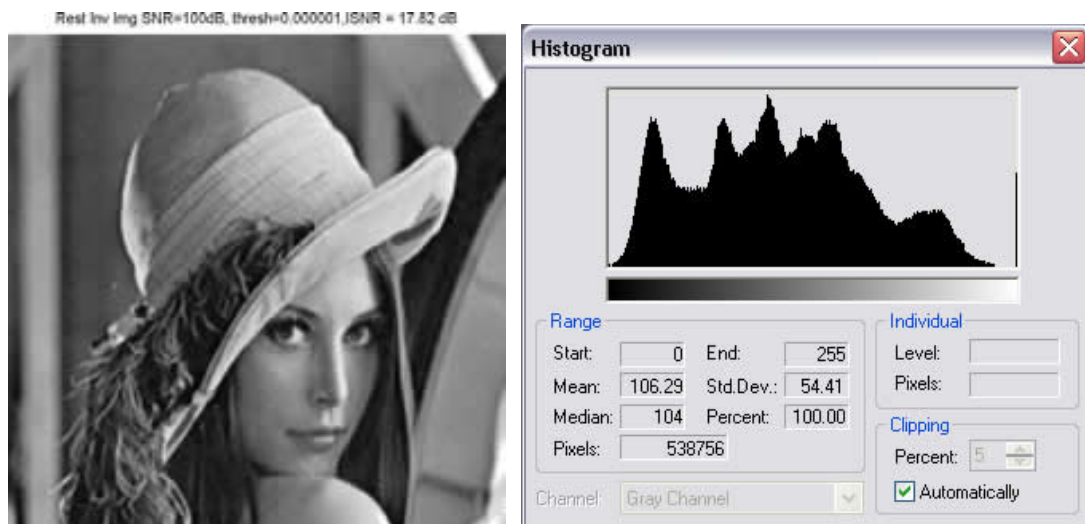


Figura C.34 – Filt. inversa, Embaçam.+ SNR=100dB, thresh=0.000001, ISNR= 17.82dB e seu histograma.

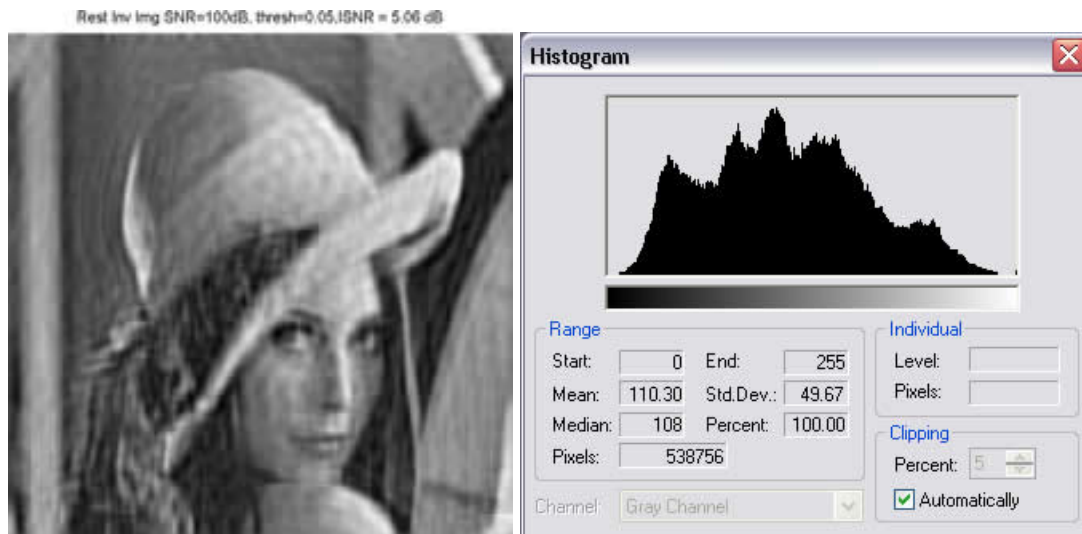


Figura C.35 – Filt. inversa, Embaçam.+ SNR=100dB, thresh=0.05, ISNR= 5.06dB e seu histograma.

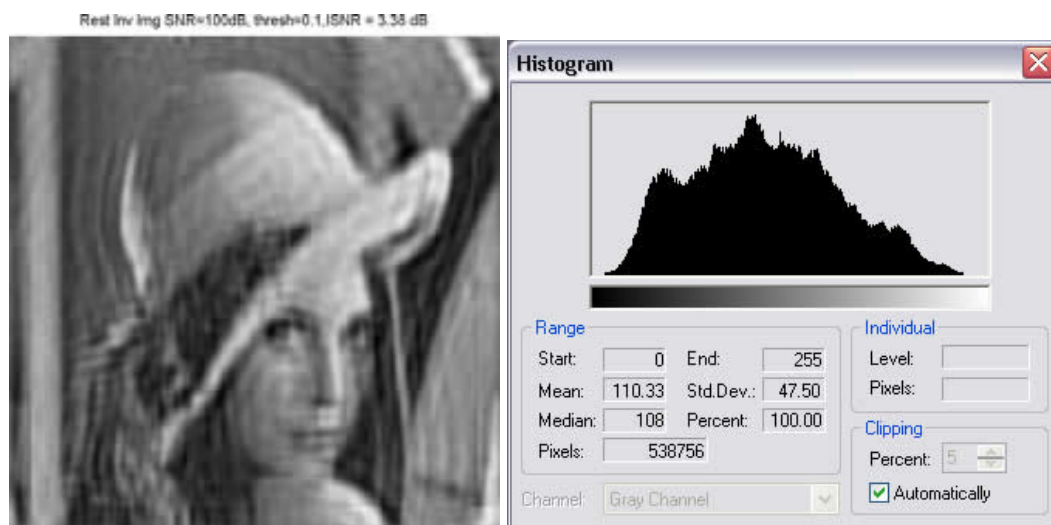


Figura C.36 – Filt. inversa, Embaçam.+ SNR=100dB, thresh=0.1, ISNR= 3.38dB e seu histograma.

ANEXO D – Filtragem Wiener

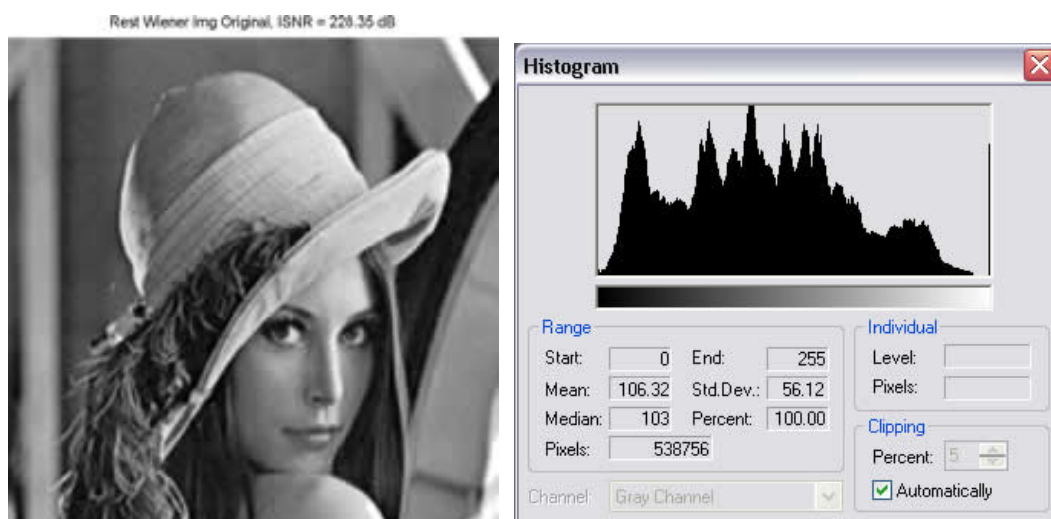


Figura D.1 – Filt. Wiener, Imagem Original, ISNR= 228.35dB e seu histograma.

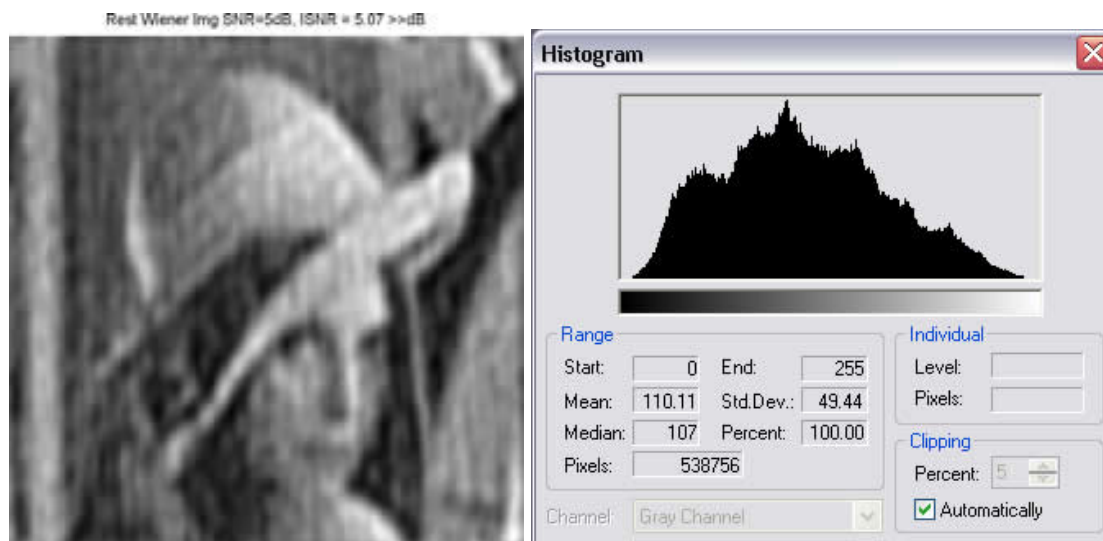


Figura D.2 – Filt. Wiener, Embaçam.+SNR=5dB, ISNR= 5.07dB e seu histograma.

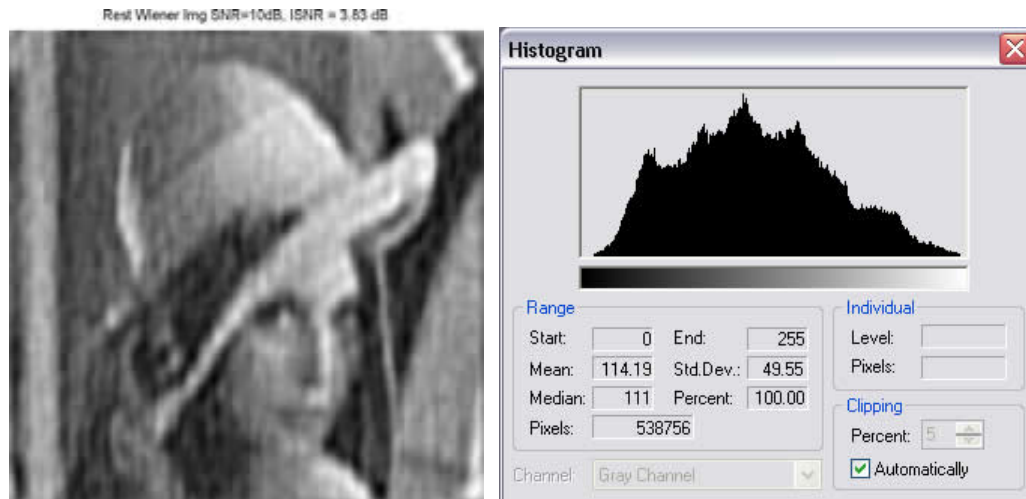


Figura D.3– Filt. Wiener, Embaçam.+SNR=10dB, ISNR= 3.83dB e seu histograma.

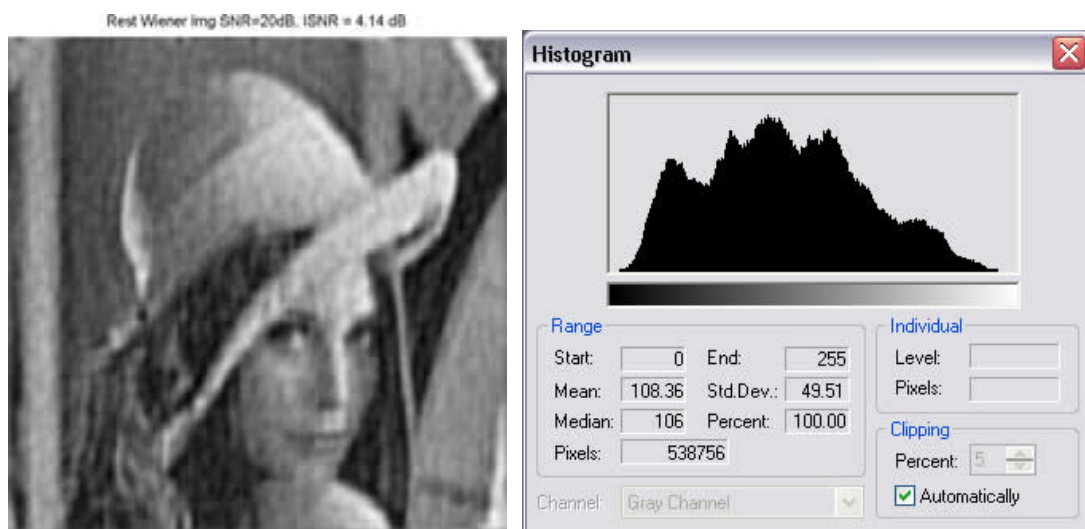


Figura D.4– Filt. Wiener, Embaçam.+SNR=20dB, ISNR= 4.14dB e seu histograma.

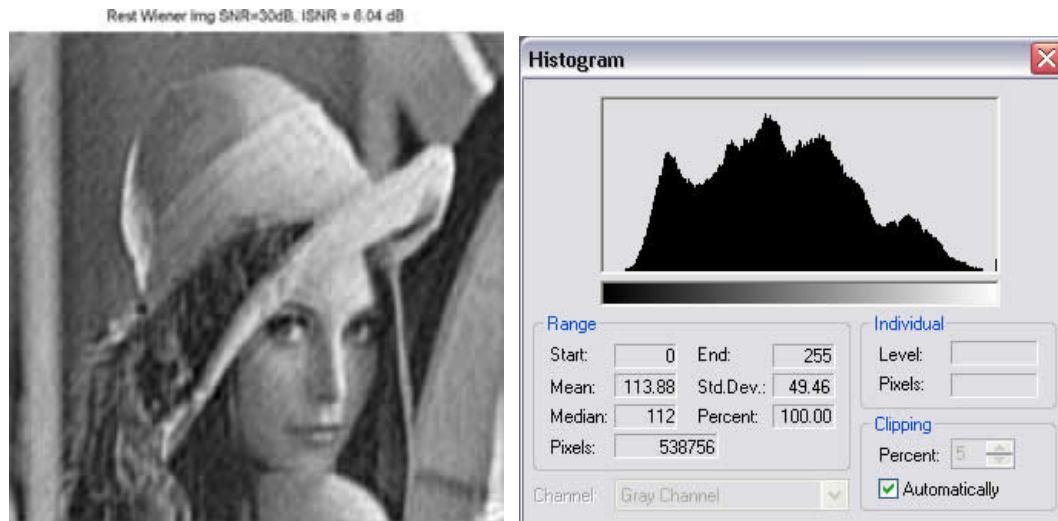


Figura D.5 – Filt. Wiener, Embaçam.+SNR=30dB, ISNR= 6.04dB e seu histograma.

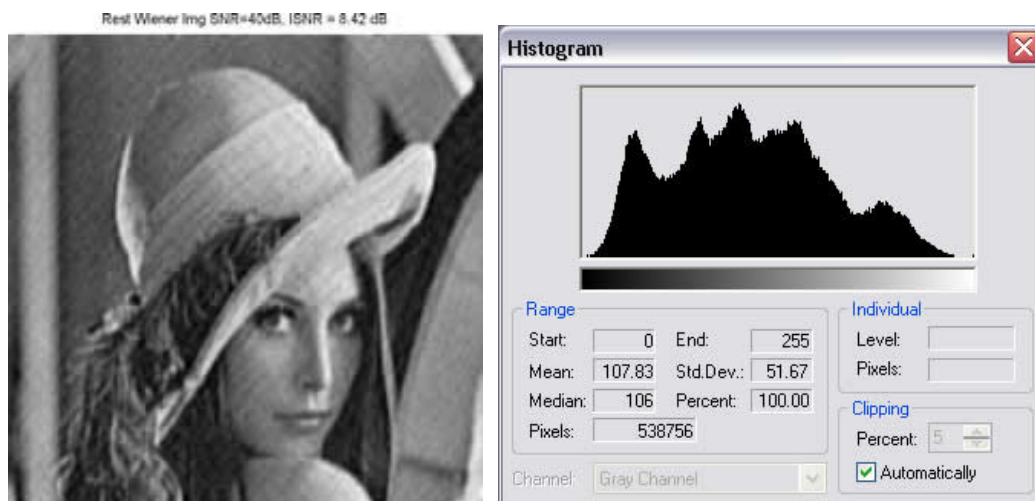


Figura D.6 – Filt. Wiener, Embaçam.+SNR=40dB, ISNR= 8.42dB e seu histograma.

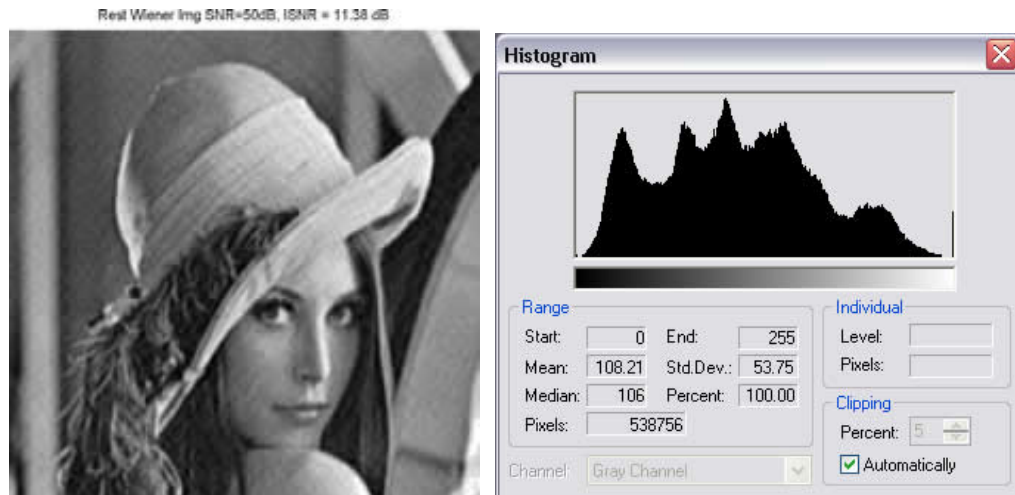


Figura D.7 – Filt. Wiener, Embaçam.+SNR=50dB, ISNR= 11.38dB e seu histograma.

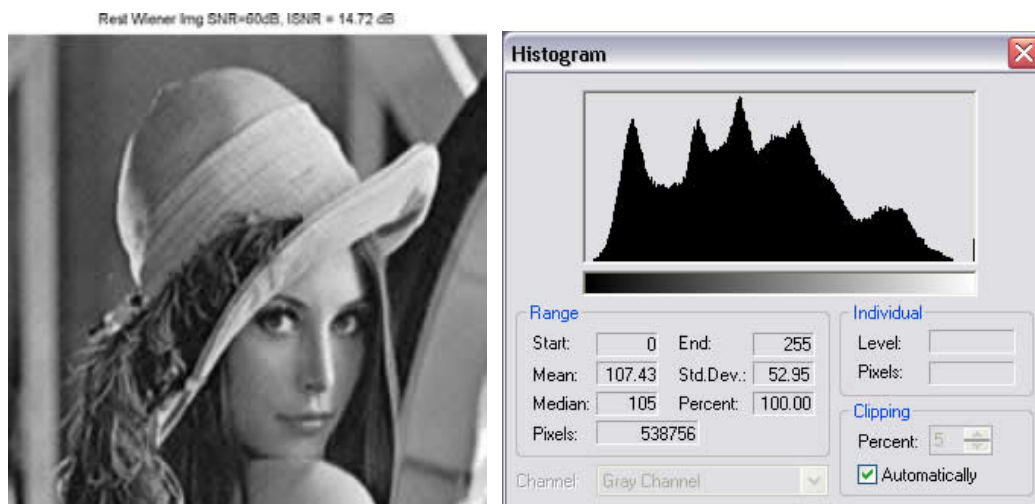


Figura D.8 – Filt. Wiener, Embaçam.+SNR=60dB, ISNR= 14.72dB e seu histograma.



Figura D.9 – Filt. Wiener, Embaçam.+SNR=70dB, ISNR= 18.47dB e seu histograma.

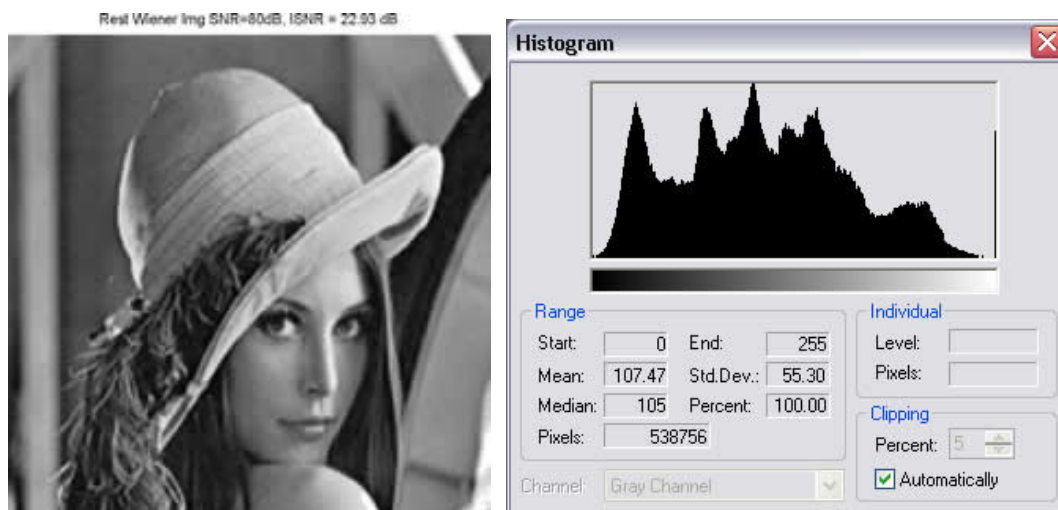


Figura D.10 – Filt. Wiener, Embaçam.+SNR=80dB, ISNR= 22.93dB e seu histograma.

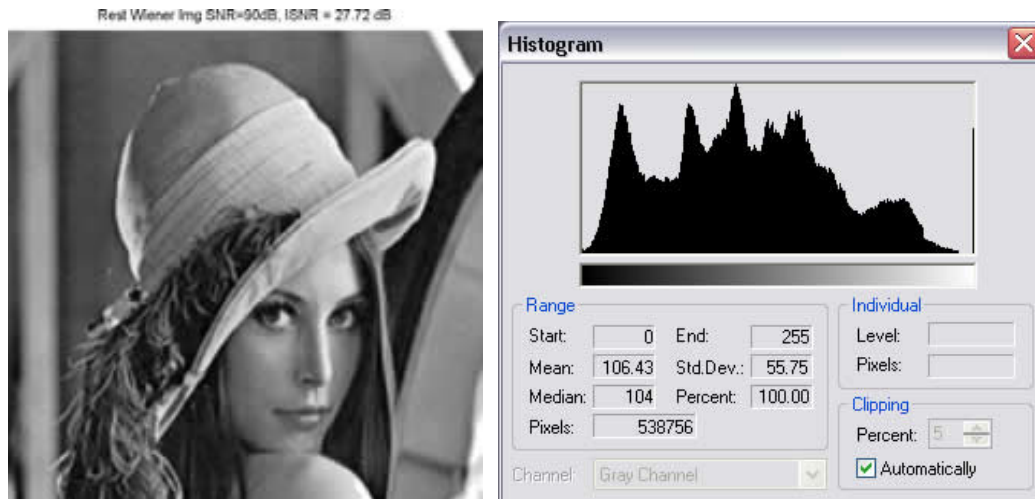


Figura D.11 – Filt. Wiener, Embaçam.+SNR=90dB, ISNR= 27.72dB e seu histograma.

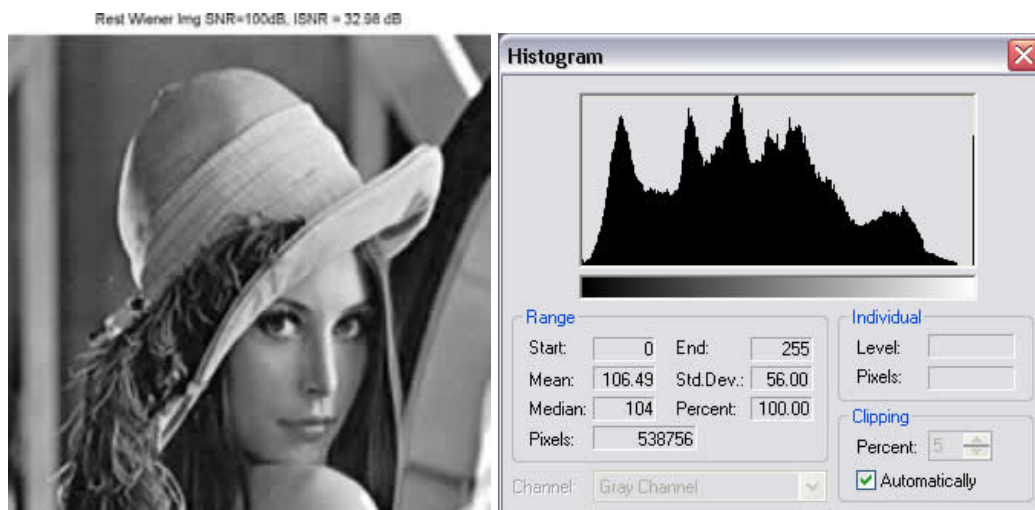


Figura D.12 – Filt. Wiener, Embaçam.+SNR=100dB, ISNR= 32.98dB e seu histograma.